Bayesian framework for building gene regulatory networks including delays

Iti Chaturvedi

2013

Iti Chaturvedi.(2013). Bayesian framework for building gene regulatory networks including delays. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/55054>

<https://doi.org/10.32657/10356/55054>

NANYANG TECHNOLOGICAL UNIVERSITY

Bayesian Framework for building Gene Regulatory Networks including Delays

A PhD Thesis

Submitted to the School of Computer Engineering
of Nanyang Technological University

by

Iti Chaturvedi

for fulfillment of the requirement
for the Degree of Doctor of Philosophy

Supervisor: Assistant Professor Vitali Zagordonov

Co-Supervisor: Associate Professor Kwoh Chee Keong,
Co-Supervisor: Professor Jagath C. Rajapakse

Oct, 2013

Contents

Abstract	v
Acknowledgments	viii
Author's Publications	ix
Abbreviation	x
Notations	xiii
List of Figures	xv
List of Tables	xviii
List of Algorithms	xx
1 Introduction	1
1.1 Motivation and Objectives	1
1.2 Major Contributions of the thesis	3
1.3 Gene Regulatory Networks	5
1.4 Methods of Building GRN	6
1.5 Preprocessing of Data	9
1.5.1 Missing Value Estimation	9
1.5.2 Noise Removal	9
1.5.3 Discretization	10
1.6 Validation	10
1.6.1 Synthetic multinomial data	10
1.6.2 Validation using Protein Interaction Data	11
1.7 Layout of the Thesis	11
2 Convergence and Stability of Building GRN with DBN	13
2.1 Motivation and Contributions	13
2.1.1 Notations used in the Thesis	15
2.2 Bayesian Networks	16
2.3 Dynamic Bayesian Networks	17
2.4 Parameter Learning	19
2.4.1 Maximum Likelihood Estimates of Parameters	19
2.4.2 Prior of Parameters	20

2.5	Structure Learning	21
2.5.1	Initialization of the structure	23
2.5.2	MCMC Simulations	23
2.5.3	Genetic Algorithm	24
2.6	Stability	25
2.6.1	Simulation of Gene Expression Time-Series	26
2.6.2	Stability of Structure Estimation	26
2.7	Existing Techniques	27
2.7.1	Multivariate Autoregressive Models	27
2.7.2	Gaussian Bayesian Networks (GBN)	29
2.8	Bootstrapping	29
2.8.1	Vector Sieve Bootstrap	30
2.8.2	Probabilistic Residual Bootstrap	31
2.8.3	Validation	32
2.9	Experiments on Synthetic Data	33
2.9.1	Structure Learning	35
2.9.2	Effects of Prior	36
2.9.3	Effects of the Number of Time points	37
2.9.4	Stability of the Networks	38
2.9.5	VAR and Gaussian Networks	39
2.9.6	Sieve and Residual Probabilistic Bootstrap	39
2.9.7	Experiments on Real Datasets	41
3	Higher-order Dynamic Bayesian Networks (HDBN)	51
3.1	Motivation and Contributions	52
3.2	Parameter estimation in HDBN	53
3.3	Variable-Order MCMC (VMCMC)	56
3.3.1	Detailed Balanced Condition	57
3.3.2	Stability of Delays	58
3.3.3	Probabilistic Residual Bootstrap	59
3.4	Experiments with Synthetic Data	61
3.4.1	Structure Learning	61
3.4.2	Effects of the Number of Time points	61
3.4.3	Stability of the Networks	63
3.4.4	VAR Networks	63
3.4.5	Vector Sieve Bootstrap	63
3.5	Discussion	68

4	Building GRN with Skip-chain Models	71
4.1	Motivation and Contributions	72
4.2	Parameter Estimation for Skip-Chain Model	74
4.2.1	Hidden Markov Models (HMM)	74
4.2.2	Dirichlet Prior for Skip-Edge	77
4.3	Experiments with Synthetic Data	78
4.3.1	Structure Learning	79
4.3.2	Effects of the Number of Time points	79
4.3.3	Stability of the Networks	80
4.3.4	VAR Networks	81
4.3.5	Vector Sieve Bootstrap	81
4.4	Discussion	85
5	Discussion and Conclusions	89
5.1	Comparison of different Models	90
5.1.1	Priors on Models	90
5.1.2	Effect of Number of Samples and Genes	91
5.1.3	Stability of the Networks	91
5.1.4	Bootstrapping Models	92
5.1.5	Validation of Real datasets	92
5.2	Discussion	93
5.2.1	Stability of Models	94
5.2.2	Higher-order delays	94
5.2.3	Long delays	95
5.2.4	Optimization	95
5.3	Limitations and Future Work	96
5.3.1	GRN framework and Implementation	97
5.3.2	Acquisition of Feature Sets	97
	References	99
A	Appendix	111
A.1	Noise Removal	111
A.2	Maximum Likelihood estimates for BN	111
A.3	Priors of Parameters	113
A.4	Boolean Networks	115
A.5	Differential Equations	116
A.5.1	Linear Differential Equation Model	116
A.5.2	Recurrent Neural Networks	118

A.5.3	Non-Linear Differential Model: S-systems	118
A.5.4	Non-Parametric Regression with Gibbs Prior	119
A.6	Fusion methods	120
A.6.1	Naive Bayes Method	120
A.6.2	Mining Cross-Graph Quasi-Cliques	122
A.6.3	Simulated Annealing and SVM score	123

Abstract

Any biological activity is due to several genes regulating one another in a gene regulatory network (GRN). GRN are causal models where activation of one gene could cause another gene to be activated or inhibited. Dynamic Bayesian networks (DBN) are being widely used to model GRN because of their ability to capture complex regulatory interactions among genes. DBN are presumed to be of first-order such that gene expression at a particular time point is only dependant on the previous time point. However, regulatory events are inherently asynchronous and transpire at different times, and involve delays. In order to account for delays in GRN, we propose higher-order DBN.

Chapter 1 gives an introduction to GRN and describes earlier works of building GRN, which are broadly classified in to (i) Boolean networks, (ii) differential equations, and (iii) stochastic models. The limitations of these approaches in representing delays in regulatory networks and the motivation to this research work are then highlighted. Four sets of benchmark data used in the demonstration of research work in the thesis are described next: synthetic data, two sets of cell-cycle data, and DNA repair data collected in *M. tuberculosis*. The preprocessing techniques used and rational of using them are demonstrated next. The first chapter ends with a summary of contributions made in this thesis work, which are detailed in Chapters from 2-5. Chapter 6 concludes the thesis with a summary and potential future works.

Chapter 2 describes the DBN approach for building GRN and introduces novel ways of evaluating their confidences, robustness, and structure learning. Bayesian networks are acyclic while DBN are able to handle feedbacks and loops. We demonstrate how Dirichlet priors improve the estimation of parameters of GRN; next we compare structural learning by Markov chain Monte Carlo (MCMC) technique and genetic algorithm (GA), and lastly how bootstrapping helps determine the confidence of structures and connections. The MCMC technique generates more accurate structures on small set of genes while GA-based structure learning is more suitable for a large number of genes. We overcome the errors due to generalization and instabilities with bootstrapping. A probabilistic approach of bootstrapping of GRN is proposed and we

compare it with sieve bootstrapping for evaluation of the confidences of GRN structures and parameters.

Pathways are inherently asynchronous with delays in gene regulation and assembly of gene products. There the first-order assumption in DBN is extended to higher-orders where expressions of genes depend on the expression of parents from r (> 1) previous time points. Only a few have considered higher-order DBN (HDBN) to represent delays in regulatory networks but the order or the maximum delay of such networks has been fixed. To overcome this limitation, in Chapter 3, we introduce variable-order DBN (VDBN). The order is automatically determined by using a novel variable-order MCMC technique. This approach rendered more accurate GRN structure because of its ability to achieve correct delays and thereby biologically more plausible networks. A method for validation of estimated networks by using protein-protein interaction (PPI) data is presented. The results with the benchmark data show improved accuracy of networks built. We investigate biological relevance of the genes, connections, and delays represented in the network.

The higher-order and variable-order DBN models are computationally intractable when the delays or orders of the networks are large. In Chapter 4, a skip-chain Markov model is introduced to handle long delays in regulatory networks. It automatically determines the optimal delays by using a hidden Markov model (HMM) and hence does not need to heuristically search the high-dimensional solution space. In this approach, we propose novel techniques to identify time-delayed interaction features and determine the optimal structure by computing the Viterbi scores and using an MCMC.

In conclusion, the knowledge of regulatory networks gives a comprehensive picture of how genes function and interact with one another to execute a specific biological function. Two approaches of building higher-order gene regulatory networks were introduced in this research - VDBN and skip-chain DBN - that provide more accurate ways of building networks than earlier methods by accurately representing delays. The gene networks derived for the benchmark datasets, the core-genes and -networks, and their biological relevance were investigated. In addition, we proposed methods to validate, and determine robustness and significances of networks derived. This research should lead to building more accurate pathways in a data driven manner combining various sources of bioinformatics data. When the delays of regulations are large, the number of parameters involved becomes intractable. Our algorithms not only targeted the tradeoff between optimality and complexity of solution landscapes but also yielded biologically relevant networks. This work can be extended to

building more complex pathways by using multiple sources of bioinformatics data.

Acknowledgments

I heartily thank my advisor, Professor Jagath C. Rajapakse , for his excellent support, guidance, and technical help right from the start of my candidature. He has given me excellent research directions and helped me in developing critical scientific thinking. Without his efforts, the road to this research would not have been so enriching, fulfilling, and smooth. I would also like to thank my co-supervisor Prof Vitali Zagordonov for reading my thesis and helping with refinements during the later part of my PhD.

I acknowledge former research fellows in my lab, particularly Dr Cheng Jierong and Dr. Piyush, for their excellent technical support and discussions to shape up my early research ideas. BioInformatics Research Centre (BIRC) gave many trustworthy friends such as Swathi, Merlin, Liu Song, Pooja, Zheng Bo, Wenting etc who made my stay at BIRC joyful and I will always cherish their memory. Special credit will go to the technical staff Asai and Sing Yau in BIRC, for their diligent assistance with the High Performance servers. I am grateful to School of Computer Engineering (SCE) and Nanyang Technological University for financially supporting me to pursue the doctoral studies . Financial supports to attend various conferences by SCE, IEEE and APBionet are much appreciated.

I express my sincere gratitude to Singapore-MIT Alliance and School of Physics and Mathematical Sciences for providing me opportunities to serve as Teaching Assistant for two consecutive years, which enhanced my skills in mathematics and statistical analysis. I thank all the researchers who have made their datasets available to predict gene networks. Without them, this thesis would never have happened. I sincerely express my thanks to my family for their continuous support and encouragement without which I would not have come so far.

Author's Publications

Journal Papers

- Chaturvedi I and Kee YH, The Power of Now: Brief Mindfulness Induction Led to Increased Randomness of Clicking Sequence. *Motor Control*. 2012 Oct;
- Chaturvedi I and Rajapakse JC, Building Gene Networks with Time-Delayed Regulations. *Pattern Recognition Letters*. Volume 31, Issue 14, 15 October 2010, Pages 2133-2137
- Chaturvedi I and Rajapakse JC, Detecting robust time-delayed regulation in *Mycobacterium tuberculosis*. *BMC Genomics*. 2009 Dec 3;10 Suppl 3:S28.

Conference Papers

- Zheng J, Chaturvedi I and Rajapakse JC, Integration of Epigenetic Data in Bayesian Network Modeling of Gene Regulatory Network. *PRIB* 2011
- Chaturvedi I and Rajapakse JC, Stability of Inferring Gene Regulatory Structure with Dynamic Bayesian Networks. *PRIB* 2011
- Chaturvedi I and Rajapakse JC, Gene Regulatory networks with variable order dynamic Bayesian networks. *IJCNN* 2010
- Chaturvedi I, Rajapakse JC, Fusion of Gene Regulatory and Protein Interaction Networks Using Skip-Chain Models. Rajapakse, J.C., Schmidt, B., Chetty, M. (Eds.), *Pattern Recognition in Bioinformatics, Lecture Notes in Bioinformatics, LNBI 5265*, (2008) 214-224
- Chaturvedi I, Sakharkar M, Rajapakse JC, Validation of Gene Regulatory Networks from Protein-Protein Interaction Data: Application to Cell-Cycle Regulation. Rajapakse, J.C., Schmidt, B., Volkert, G. (Eds.), *Pattern Recognition in Bioinformatics, Lecture Notes in Bioinformatics, LNBI 4774*, (2007) 300-310

Abbreviations

DNA	Deoxyribonucleic acid
RNA	Ribonucleic acid
GRN	Gene Regulatory Network
BN	Bayesian Network
DBN	Dynamic Bayesian Network
HDBN	Higher Order Dynamic Bayesian Network
VDBN	Variable-order DBN
PPI	Protein Protein Interaction
PPIN	Protein Protein Interaction Network
MVAR	Multivariate Autoregressive
HMM	Hidden Markov Model
VA	Viterbi Algorithm
SOM	Self-Organizing Maps
PCA	Principal Component Analysis
ICA	Independent component analysis
ODE	Ordinary Differential Equations
LDE	Linear Differential Equation
SVM	Support Vector Machine
KNN	K Nearest Neighbors
MTB	Mycobacterium Tuberculosis

JADE	Joint approximate Diagonalization of Eigenmatrices
S-systems	Stochastic systems
BDe	Bayesian Dirichlet equivalent
BIC	Bayesian Information Criterion
MCMC	Markov Chain Monte Carlo
MDL	Minimum Description Length
MI	Mutual Information
RSS	Residual Sum of Square Errors
RNN	Recurrent Neural networks
MSE	Mean Squared error
ML	Maximum Likelihood
CPD	Conditional Probability Distributions
GA	Genetic Algorithm
BNRC	Bayesian Non Parametric Regression Criterion
EM	Expectation Maximization
NB	Naive Bayes
DAG	Directed Acyclic Graphs
GBN	Gaussian Bayesian Networks
tdGRN	time-delayed Gene Regulatory Networks
F-measure	Fisher Measure
TP	True Positives
FP	False Positives
FN	False Negatives
TN	True Negatives
GEO	Gene Expression Omnibus

NCBI	National Center for Biotechnology Information
RJMCMC	Reversible Jump MCMC
VMCMC	Variable-Order MCMC
MAP	Maximum A-posteriori
BioGRID	Biological General Repository for Interaction Datasets
MIPS	Mammalian Protein-Protein Interaction Database
CRF	Conditional Random field
REVEAL	Reverse Engineering Algorithm
KEGG	Kyoto Encyclopedia of Genes and Genomes
LDE	Linear Differential Equation
FT	Fourier transform
STEM	Short time series expression miner
BFGS	Broyden-Fletcher-Goldfarb-Shanno
COG	Clusters of orthologous proteins

Notations

G	Set of genes in graph
Γ	Set of discrete states of a gene
s	Structure of Bayesian network
n	Number of Genes in a network
θ	Parameters of the Bayesian network
θ_{ijk}	Parameter for state k of gene i with parent state combination j
N_{ijk}	Number of instances of θ_{ijk} in the data
q	Prior probability for number of parents
$p(s x)$	Probability of structure s given data x
$q(s x)$	Prior probability of structure s given data x
β	Regression co-efficient for the network
χ^2	Chi-square Test
Σ	Covariance matrix
Σ_{ia_i}	Covariance matrix for parent set a_i
μ	Mean
N	Normal Distribution
ϵ_{ij}	Probability of errors for gene i and parent state j
M_{ijk}	Error counts for gene i with parent state combination j and gene state k
p	Confidence level of Bootstrap
$\underline{\delta}$	Lower interval for Bootstrap

$\bar{\delta}$	Upper interval for Bootstrap
$\rho(c^b, c^{b'})$	Hamming distance between two networks
$h(y_{t-k}, y_t)$	k order edge using PPI
$x_i(t)$	Expression level of gene i at time t
T	Total length of time-series
r	Markov order
γ_+	Probability of birth step in VMCMC
γ_-	Probability of death step in VMCMC
γ_0	Probability of no change in VMCMC
$z(r)$	Number of delays with mutual information below threshold
u	Iteration of MCMC
B	Number of bootstrap samples
s^b	b^{th} sample in bootstrap
$\rho(c^b, c^{b'})_{d_1, d_2}$	Hamming distance networks with delay d_1 equivalent to d_2
θ_{ijzk}	Parameters for delay combination z
α_{ijzk}	Pseudo count for delay combination z
s_t	Starting time-point of skip-edge
t	Ending time-point of skip-edge
$y_{s_t:t}$	Labels of gene at in skip path from s_t to t
$a_{l,m}$	Transition probability from gene i to gene j
$b_l(k)$	Emission probability for state k for gene i
$\delta_m(t)$	MAP path to gene m by Viterbi algorithm
$h(x_i, a_i, s_t, t)$	Probability of skip-edge
$f(x_i, a_{i(t-r:t)}, t)$	Linear probability of HDBN
$h'(x_i)$	Total probability of skip-edge

λ	Prior of Viterbi algorithm
α	Interpolation factor for weighing skip-edge specified by user

List of Figures

1.1	Illustration of (a) the traditional pathway model and (b) the regulatory network model. The traditional pathway models are only cascades of interactions while network models can represent all possible interactions.	6
1.2	Yeast cell cycle phase G1-S, it can be seen that the transcriptional factor Myc activates many different genes.	7
1.3	Expression levels for genes in <i>M.tuberculosis</i> dna repair pathway before and after noise removal (a) 4 replicates for time point at 2h after dna damage; and (b) First independent component of signals.	10
2.1	Illustration of a gene regulatory network represented in a Bayesian network. Nodes represent genes and connections represent causal interactions: x_i represents the activity of gene i and $p(x_i x_j)$ represents the regulation or the conditional probability of gene i having state x_i when gene j has the state x_j	16
2.2	Transition network of a gene regulatory network of n genes , represented by a dynamic Bayesian network (DBN)	18
2.3	Illustration of convergence of the structure with (a) MCMC simulations and (b) GA for a network of 100 genes and 100 time samples	36
2.4	Effect of varying hyper-parameters for a network of 100 genes and 100 time samples on F-measure and Stability against (a) Dirichlet pseudo counts with $q = 0.2$ (b) Structural prior on average number of parents q with $\alpha = 1$	37
2.5	Distribution of stability of edges of gene regulatory networks obtained using MCMC and GA (a) with 50 genes and 50 time samples and (b) with 100 genes and 100 time samples	38
2.6	Target network of 9 genes in the cell cycle dataset of Chou et. al	43
2.7	Target network of 9 genes in MTB dna repair pathway	45

2.8	Time-delayed interactions in a predicted network using DBN bootstrap in yeast cell cycle.	48
2.9	Time-delayed interactions in a predicted network using DBN bootstrap in <i>M.tuberculosis</i> DNA repair pathway.	49
2.10	Time-delayed interactions in a predicted network using DBN bootstrap in Hela human cell cycle.	50
3.1	Representation of second-order DBN and its connectivity matrix.	54
3.2	Time-delayed interactions in a predicted network using HDBN bootstrap in yeast cell cycle.	67
3.3	Time-delayed interactions in a predicted network using HDBN bootstrap in <i>M.tuberculosis</i> DNA repair pathway.	68
3.4	Time-delayed interactions in a predicted network using HDBN bootstrap in Hela human cell cycle.	69
4.1	State transition diagram illustrating six time points and four genes in a DBN. The states $\{-1, 0, 1\}$ represent down-, un-, and up-regulated genes. The dashed edges are linear $r = 1, 2$ order edges found by linear features. The solid edge represents a skip-edge over four time points.	76
4.2	A skip chain model has overlapping skip-edges which model long-distant dependencies.	78
4.3	Distribution of error probabilities of predicted edges of regulatory networks obtained using VMCMC(a) Chou <i>et. al.</i> (9 genes) having up-to five delays (b) Tuberculosis (9 genes) having up-to four time point delays (c) Hela (51 genes) having up-to seven time point delays.	83
4.4	Time-delayed interactions in a predicted network using skip-chain DBN bootstrap in yeast cell cycle.	85
4.5	Time-delayed interactions in a predicted network using skip-chain DBN bootstrap in <i>M.tuberculosis</i> DNA repair pathway. .	86
4.6	Time-delayed interactions in a predicted network using skip-chain DBN bootstrap in Hela human cell cycle.	87

List of Tables

2.1	Accuracy of building DBN-MCMC model with Dynamic Bayesian Networks in the presence of 10% Gaussian noise.	35
2.2	Accuracy of derived networks derived with a stability threshold of 0.5 by using MCMC simulation and GA	39
2.3	Accuracy of networks derived using DBN, VAR and Gaussian BN for 10 genes when synthetic data is generated from the model. The running time in mins is also reported.	40
2.4	Accuracy of networks derived using DBN and Gaussian BN for 10 genes and 20 genes when continuous synthetic time series of length 200 was used.	40
2.5	Accuracy of vector sieve and residual bootstrap for 30 time points	40
2.6	First- and higher-order regulations predicted by DBN(d) and VAR models built on datasets by Chou <i>et. al.</i> (9 genes), Tuberculosis (9 genes) and Hela cell line(51 genes). The number of genes in the dataset, (likelihood), total number of regulations, true positives with protein-protein interaction data. $\hat{TP}$ are edges predicted with incorrect direction and k -skip validation is using gene cascades of k genes.	47
2.7	Top 9 hubs obtained for 51 genes in Hela cell line of human cell-cycle, ranked in the order of their connectivity. The hubs obtained from the BioGrid are also given for comparison.	49
3.1	Accuracy of networks derived using VMCMC and MCMC when five time delays are considered on 100 genes and 100 time points	62
3.2	Accuracy of building HDBN-MCMC model with Dynamic Bayesian Networks in the presence of 10% Gaussian noise.	63
3.3	Stability between delays of length zero, one, two and three when building HDBN-MCMC model with Dynamic Bayesian Networks in the presence of 10% Gaussian noise.	64

3.4	Accuracy of networks derived using VAR and HDBN when three time delays are considered on 10 genes and 100 time points . . .	64
3.5	Accuracy of vector sieve and residual bootstrap for 50 time points and upto three time point delays	65
3.6	First- and higher-order regulations predicted by DBN(d), HDBN(h), and skip-chain(d:h) models built on datasets by Chou <i>et. al.</i> (9 genes), Tuberculosis (9 genes) and Hela cell line(51 genes). The number of genes in the dataset, (likelihood), total number of regulations, true positives with protein-protein interaction data. $\hat{TP}$ are edges predicted with incorrect direction and k -skip validation is using gene cascades of k genes.	66
3.7	Top 9 hubs obtained for 51 genes in Hela cell line of human cell-cycle, ranked in the order of their connectivity. The hubs obtained from the BioGrid are also given for comparison. . . .	69
4.1	Accuracy of networks derived using prior for skip-edges and no-prior when five time delays are considered on 100 genes and 50 time points	80
4.2	Accuracy of building skip-chain MCMC model with Dynamic Bayesian Networks in the presence of 10% Gaussian noise. . . .	81
4.3	Stability between delays of length zero, one, two and three when building HDBN-MCMC model with Dynamic Bayesian Networks in the presence of 10% Gaussian noise.	82
4.4	Accuracy of networks derived using VAR and HDBN when five time delays are considered on 10 genes and 100 time points . . .	82
4.5	Accuracy of residual bootstrap for 50 time points and up-to five time point delays	83
4.6	First- and higher-order regulations predicted by DBN(d), HDBN(h), and skip-chain(d:h) models built on datasets by Chou <i>et. al.</i> (9 genes), Tuberculosis (9 genes) and Hela cell line(51 genes). The number of genes in the dataset, (likelihood), total number of regulations, true positives with protein-protein interaction data. $\hat{TP}$ are edges predicted with incorrect direction and k -skip validation is using gene cascades of k genes.	84
4.7	Top 9 hubs obtained for 51 genes in Hela cell line of human cell-cycle, ranked in the order of their connectivity. The hubs obtained from the BioGrid are also given for comparison. . . .	85

List of Algorithms

1	Markov Chain Monte Carlo	24
2	Genetic Algorithm for Structure Learning	25
3	Generating samples of gene expression data	26
4	Bootstrapping time series	32
5	k -skip Validation	34
6	Variable-order DBN	58
7	Bootstrapping time series	60

Chapter 1

Introduction

DNA replication, chromosome segregation, and mitosis are fundamental and periodic events in a cell. Precise co-ordination of unidirectional transitions between different stages is critical to cell integrity and survival. Progression of such events is regulated and accompanied by periodic fluctuations in the expression levels of numerous genes and their regulatory actions. A genome-wide characterization of mRNA transcript levels is hence needed for understanding cellular function and biological activity in general.

Though there is a plethora of techniques for building gene regulatory networks, each method has its own pros and cons. In particular, previously the delay in regulation has not been discussed in relation to building regulatory networks. This thesis considers incorporating delays of regulation, including very long delays in interactions that represent feedbacks and regulations in real pathways. Furthermore, bootstrapping and priors are also used to regularize or compensate for the lack of enough time points for building models. Lastly, validation is done using synthetic data and corresponding protein interactions reported in BioGrid [1].

This chapter gives an introduction to the research carried out throughout this thesis. First motivations of this research are highlighted and then the major contributions of the work are stated. Thereafter, background work that leads to this research is given. A detailed review of the work related to each chapter is given. For validation of the network both synthetic and real data has been used.

1.1 Motivation and Objectives

Genes are the functional units of DNA and code for proteins. Usually a group of genes are involved in a specific biological activity. Gene regulatory networks

(GRN) represent interactions and regulations among a group of genes that are involved in a particular biological process. The GRN are causal models where regulatory activities of genes represent how one gene causes another gene to be activated or inhibited. There have been a plethora of techniques in the literature describing different techniques of building gene regulatory networks [2]. Dynamic Bayesian networks (DBN) are popular among other methods because of their ability to infer causal inferences of genetic networks [3]. The applications of such techniques to gene expression time-series observations find near optimal interaction networks [4]. However, DBN presume first-order interactions that gene expression at a particular time point only depends on the previous time point. The gene regulatory events are inherently asynchronous and transpire at different times. However, delays of gene regulatory events have thus far received little attention when building gene regulatory networks from gene expression data.

This thesis investigates how delays are represented in GRN and proposes several schemes based on higher-order Bayesian networks to represent delays in gene regulation. Though dynamic Bayesian networks (DBN) have been previously proposed to build gene regulatory networks, they have not been explored in higher-order to represent varying delays of regulation [5], [6]. Techniques to handle delays are confounded due to the presence of feedbacks that span over varying durations among genes across a pathway. Since exact computation of marginals in graphs with loops does not exist, a reparameterization of higher-order DBN is necessary. To account for delays, we extend Bayesian networks to higher-order in order to handle delays in gene regulatory networks. However, when the delays increase, the number of parameters in the network exponentially increase. It is necessary to handle the delays of the networks in order to infer underlying mechanism of gene regulation. Very few authors in the past have worked on finding delays in gene regulation [7], [8].

Microarray data is inherently noisy due to imperfections on slides and errors in fabrication. Microarray experiments are very expensive and data can be gathered only on a few time points to make inferences underlying most experiments. These techniques to handle noise and artifacts in microarray data and their effects on data analysis have not been explored comprehensively. Therefore, in order to mitigate the effects of limited number of samples, we investigate the use of priors. Bootstrapping is also proposed as another way of assessing confidences of edges in the presence of generalization errors and temporal aggregation [9].

Validation of our methods are performed using synthetic data. The utility of our techniques are demonstrated on real data. For real networks, we attempt to validate the GRN by using data available in the protein-protein interaction (PPI) databases. Validation of delays in the higher-order model remains a challenging task and has been earlier done using biological or phase information [10]. But such methods are expensive, tedious and have technical limitations. Therefore in-silico methods that determine delays of gene expression networks are necessary. We also analysed core hubs that are significantly present in the network and compare with those reported in literature.

1.2 Major Contributions of the thesis

In the pursuit of building GRN with delays, we propose variants of higher-order DBN in chapters 2, 3 and 4. Chapter 2 describes techniques to evaluate the stability and robustness properties of the networks; Chapter 3 describes a variable order networks to handle variable delays and Chapter 4 uses a skip-chain model to predict very long delays. The specific contributions of this thesis are as follows:

1. The stability of DBN was important as it ensures reproducibility of networks [11]. Vector autoregressive (VAR) bootstrap have been previously used for time-series data which can be sampled from an empirical distribution of innovations or residual as extracted from the given data with an optimal autoregressive order. Since VAR needs several time-samples even for small number of genes and is applicable only on linear networks we propose a modified probabilistic bootstrap which samples using a multinomial distribution.

2. The first-order assumption in DBN leads to GRN with synchronous interactions and does not represent delays of regulation that are biologically plausible. The order of the DBN assumes that the expressions of genes are dependent on expressions of genes from a number of previous time points [10]. The parameter learning in higher-order dynamic Bayesian networks (HDBN) decomposes each state of the parent set of genes into a product of different delay states. We introduce a variable-order HDBN to automatically determine the delays of regulations.

The classical MCMC technique is impractical for determining the optimal structure of HDBN because of the computational complexity of processing the number of connections required in each cycle of iteration. Therefore, we propose a variable-order MCMC algorithm where the connectivity matrix denoting

delays between genes is learned from the data [12]. This approach renders a more accurate GRN structure because it automatically finds the optimal delay.

3. Higher-order Bayesian networks become computationally intractable at very high-orders because of the increasing number of parameters with delays. Therefore, we introduce a skip-chain model [13] that is able to handle long term delays in regulatory interactions. It automatically determines optimal delays using hidden Markov model (HMM's) and hence does not need to heuristically search the high dimensional solution space. The skip model is represented by two types of features: (a) linear features modeling the short-delays; and (b). skip features modeling long-delays. The scoring metric is a weighted sum of the two types of feature where the weight is heuristically determined.

In our model, skip features are modeled by using a HMM where the log likelihood of the network is decomposed into a sum of consecutive pairs of genes. So the maximum likelihood estimate of regulation is found by using the Viterbi algorithm (VA). Our approach consists of two stages: (a) identification of time-delayed interaction features and computation of Viterbi scores; and (b) prediction of optimal GRN by using a genetic algorithm.

4. In order to avoid over-fitting of the proposed models, we investigated two types of prior over the structure and the parameters, respectively. Structural priors reflect our prior knowledge about the average number of parents such as binomial distribution for the mean number of parents. In addition, the Dirichlet conjugate prior is used over the parameters. It is demonstrated that the introduction of priors improves precision.

5. The validation of the networks is done by generating synthetic data. Multinomial time-series are sampled from the model network adhering to power law. In case of real datasets as the ground truths of delays of the regulatory networks are not known widely, we attempt to validate the GRN by using data available in the corresponding protein-protein interaction (PPI) databases. Proteins are the expressed forms of genes, methods such as yeast-two hybrid can validate whether pairs of proteins will combine.

6. Our techniques are demonstrated on three real datasets. Two datasets obtained in yeast cell cycle and *M.tuberculosis*. Our experiments show that the present technique yields regulatory networks that better fit the gene expression data and are biologically more plausible than DBN. The sensitivity of the model is maintained even when the number of genes is very large such as human Hela cell cycle dataset of over fifty genes from cancer pathway. The model is also able to correctly identify the differential expression of genes with time resulting in feedback loops and cascaded signals.

7. Gene regulatory networks generated by our methods have numerous applications. They can be used to compare diseased and control groups. The network score is a measure of robustness of the pathway to time and can be used to cluster patients and healthy people. Another significant application is the study of hub genes in biological pathways; any knock-out of hub proteins can significantly alter the activities of the system. Hence our approach can be used to study the effects of such knock-outs visually.

Since, real pathways have inherent delays we hope that by using the additional parameter r to model delays we can predict more realistic and hence accurate networks from time series data. Biological datasets are markedly different from other time-series as the number of genes is very large > 1000 and the number of time-samples is small maybe 10 or 20 due to expensive measurement techniques. By using a Dirichlet prior over parameters or a Gibbs prior over structure we can overcome these challenges. Bootstrapping can further assess the confidence of the predicted interactions.

Our networks have over 60% accuracy on benchmark pathways and can be experimentally validated. The skip-chain model takes form of a deep architecture where simple networks predicted in first layer are combined in second layer to model more complex networks making them scalable to very large sizes.

1.3 Gene Regulatory Networks

Microarray technology has enabled us to gather expressions of thousands of genes simultaneously in an unprecedented way. If collected over time or many experiments, the underlying gene regulatory networks can be constructed and inferences on underlying gene regulations be made.

Expression of a gene results in a sequence of states of biological processes, which creates a trajectory in the state space. Since the number of states of a network of n genes having d distinct expression levels is finite (d^n), all trajectories eventually end up in an 'attractor' or steady state. Attractors are analogous to "target" of an organism : for example; the end of development of a particular cell when it repairs a tissue following a response to an injury, or an adaption of metabolic gene expression following a change in the nutrient environment of a bacteria.

The correct representation is very critical for modeling GRN from gene expression data. The traditional 'pathway model' is mostly linear as illustrated in Fig. 1.1(a); The assumption that only a single gene regulates others in each

Figure 1.1: Illustration of (a) the traditional pathway model and (b) the regulatory network model. The traditional pathway models are only cascades of interactions while network models can represent all possible interactions.

subset of genes places a very strong constraint. This representation therefore cannot accommodate the intense crosstalk between nominal pathways because of their distributed nature in cell fate regulation.

Cross communication between pathways can be broadly classified into two forms: convergence and pleiotropy. For example, Myc transcriptional factor in G1-S phase of yeast cell-cycle is known to activate at least 27 genes, hence the pathway is extremely complex with a lots of crosstalk as shown in Fig. 1.2. Multivariate linear models given in Fig. 1.1 (b) however seek to explain the covariation among different genes by examining the common interacting genes. The regulatory network model decomposes the observed time-series expressions into expression of each gene given those of a few parent genes. This allows testing specific hypothesis relevant to the biological processes. For example, a network model would give a better fit than a pathway model if a complex of two genes regulates another gene while another complex of genes regulates the same gene.

1.4 Methods of Building GRN

Early approaches to building GRN assumed that each gene belongs to a single cluster of genes having similar functions. However, subnetworks do not

Figure 1.2: Yeast cell cycle phase G1-S, it can be seen that the transcriptional factor Myc activates many different genes.

provide for regulatory interactions and clustering only provides genes that are regulated in a similar manner. For example, clustering-based methods such as hierarchical clustering [14] and self-organized maps (SOM) [15], and statistical methods for computing low-dimensional or hidden representations of data such as principal component analysis have limited utility (PCA) [16] and independent component analysis (ICA) [17] could only find genes that are co-expressed. Obviously, such methods have limited utility in representing interactions or regulations among genes.

The methods of deriving GRN from gene expression time-series can be broadly classified into four categories: networks built by using (a) boolean rules [18], [19]; (b) dynamic differential equations [20], [21], [22]; (c) stochastic modeling [23], [24] and (d) fusion methods combining microarray with protein interaction data [25], [26] and [27].

Boolean networks are the simplest, but are not causal and built upon mutual information among nodes. Bayesian networks (BN) have been introduced for building gene regulatory networks in the stochastic framework [24]. BN are capable of representing causality among genes and therefore are more biologically plausible and accurate than Boolean networks. Ordinary differential equations (ODE) can model the complex regulatory dynamics. Such methods often give detailed dynamics of interactions but result in large scale linear dynamical systems. However, such modeling is possible only when the connectivity of the structure is known. Complex regulatory dynamics are often

simplified by knowing the underlying structure and the underlying structure determined by BN could be used for this purpose.

The DNA in a cell contains genes which are converted to mRNA (expressed genes) through transcription and then translated into proteins. Consequently, signal transduction pathways are often interpreted in terms of gene regulatory networks (GRN) as well as protein-protein interaction networks (PPIN). High throughput techniques allow generation of both gene and protein interaction data simultaneously. Studies that use both gene and protein expressions have been mostly devoted to a single type of data while the other type of data is restricted to validation. Using a single data source of interactions has its own limitations and could create errors in the analysis of the interactome which is the whole set of molecular interactions in a particular cell. This is due to two main reasons: firstly, both microarray and PPI data have a lot of noise due to measurement errors, varied transcriptional response in the cell and inter-functional phenomena. Secondly, complex formation and other critical interactions that regulate biological processes take place at the protein level. A protein being a product of a gene, a joint mining of gene regulatory networks and protein interaction networks could reveal genes that are co-expressed and their proteins also interact. Clusters and interactions found by joint mining will be more reliable than those found by using only one type of data. We may thereby have high confidence of finding gene clusters that are regulated by the same mechanism and belong to the same biological process.

Fusion of GRN and PPIN has been attempted by using Naive Bayes where genes are partitioned into different pathways [25]. Likelihood of the data becomes higher when genes in the same pathway have similar expression profiles and are interacting. The unified model is learned using Expectation Maximization algorithm. This method however requires the user to specify the number of pathways which is often unknown. Cross-graph mining [26] can achieve integration by looking for partial cliques in both GRN and PPIN simultaneously. A weighted score of support vector machine (SVM) classification of gene expression and functional classification of PPI was suggested in [27]. The weights are determined by simulated annealing. Prior modeling of PPIN into Bayesian learning has been done using Gibbs distribution [20]. A Gibbs random field equivalent to a first-order Markov random field is used to represent the prior graph. This approach is insufficient because many time-delayed interactions are known to exist. We have explained these methods in the Appendix A.6 .

1.5 Preprocessing of Data

Preprocessing of microarray data is necessary in order to estimate missing values, remove noise and for normalization and discretization of data. This section describes the preprocessing steps used in gene expression data for building GRN.

1.5.1 Missing Value Estimation

A key problem in building GRN is the lack of sufficient time-points in gene expression data. The raw gene expressions usually contain missing data values. This is due to insufficient resolution and corruption of images, or simply due to dust or scratches on microarray slides. Common methods to impute missing values include filling in least square estimates, iterative analysis of variance [28], randomized inference methods, and likelihood-based approaches [29]. However, these methods are model-based and lack sensitivity and robustness needed for microarray data.

In this research, KNN-impute (K Nearest Neighbors) method was used to predict missing microarray expression levels [30]. The KNN-based method selects genes with expression profiles similar to the gene of interest to impute missing values. For a missing value in an experiment of gene expressions, the algorithm finds k other genes that have an expression value most similar to the missing value in the rest of the experiments and then estimates it by the weighted average. To find k -nearest neighbours, the Euclidean distance was used as the distance criteria similar to Stanford Microarray Database (SMD) which contains raw and normalized data from microarray experiments for use in research [31].

1.5.2 Noise Removal

Since microarray data are often confounded by artifacts, array calibration, and random noise replicates of each measurement are usually available. Noise core removed from these repeated measurements by taking an average measurements. Other noise filters such as principal component analysis (PCA) [32] and independent component analysis (ICA) [33] can separate noise as insignificant components. JADE (Joint approximate Diagonalization of Eigenmatrices) algorithm is an extension of ICA which uses higher-order time delays and hence more suitable for time-series data. The JADE algorithm is a blind source separation technique that predicts the true underlying time-series by joint diagonalization of covariance matrices at different delays [33]. It yields the true

Figure 1.3: Expression levels for genes in *M.tuberculosis* dna repair pathway before and after noise removal (a) 4 replicates for time point at 2h after dna damage; and (b) First independent component of signals.

expression patterns that are independent of other source signals and noise, which may be interpreted as potential regulation pathways (Appendix A.1).

For eg. in Fig. 1.3 (a) we have used JADE on 4 replicated in MTB data at 2 hrs from start of experiment for 9 genes. The corresponding noise free first component is shown in Fig. 1.3 (b) and can be used for predicting GRN.

1.5.3 Discretization

The discretization of gene expression is required, which converts all continuous gene expressions to categorical form such as using integers. Unsupervised discretization simply divides the profile into equal bins and is not sensitive to relations among the genes. The genes were assumed to take three states $\Gamma = \{-1, 0, +1\}$ representing down-, no-, and up-regulation of genes.

1.6 Validation

To compare the performance of our models of GRN with existing ones in the literature and validate them with known biological pathways, the following procedures were used.

1.6.1 Synthetic multinomial data

The GRN in a known connectivity matrix c was synthetically generated by randomly initializing the probability distribution tables: $\{\theta_{ijk}\}_{i=1, j=1, k=1}^{n, q_i, d}$. The genes were assumed to take three states $\Gamma = \{-1, 0, +1\}$ representing down-, no-, and up-regulation of genes. Genes with no parents were set to up-

regulation with corresponding $CPD = \{0.1, 0.1, 0.8\}$, for all remaining genes the state was determined by the dominant state in the parent set.

1.6.2 Validation using Protein Interaction Data

DNA in a cell contains genes which are converted to mRNA through transcription and then translated into proteins. High throughput techniques allow generation of both gene and protein interaction data simultaneously. Studies that use both gene and protein expressions have been mostly devoted to a single type of data while the other type of data is restricted to validation [27], [25], [26]. We show validation of our results by employing BioGrid protein-protein interaction (PPI) data [34]. We also considered a k -skip validation for higher delay elements as explained in Chapter 2. Lastly, we also examined core hubs, complexes and motifs of the predicted network so that their functionality are aligned with those explained in literature.

1.7 Layout of the Thesis

The thesis is organized into four more Chapters. Chapter 2-4 describe the major work done in the thesis while chapter 5 concludes the thesis.

Chapter 2 describes how dynamic Bayesian networks (DBN) [35] are used in building GRN. Structural learning using MCMC and GA is also introduced. Next, we explain the VAR bootstrap and introduces probabilistic bootstrap for time series in order to find the confidence of connections. We demonstrate our technique by using synthetic data as well as real data obtained in cell cycle and *M.tuberculosis*. Our experiments show that the networks obtained by bootstrap yields regulatory networks that better fit the gene expression data and are biologically more plausible than DBN. The networks are validated by using protein-protein interaction (PPI) data.

Chapter 3 explains parameter learning in variable-order DBN (VDBN). The order and the optimal structure of which are achieved by using a variable-order Markov chain Monte Carlo (MCMC) method [12]. In order to validate higher delay interactions we explain the k -skip algorithm that look at the cascades of genes in the GRN, which corresponds to interactions in the PPI network [36].

Chapter 4, details the skip model are presented [13] which can handle long term delays in regulatory interactions. First we describe identification of time-delayed interaction features; followed by computation of Viterbi scores; and lastly prediction of the optimal GRN by using a genetic algorithm. Then use of priors on to avoid over-fitting is addressed.

Chapter 5 summarizes the contributions made in the present program of research, out-lines directions for future research and provides conclusions.

Chapter 2

Convergence and Stability of Building GRN with DBN

Though many methods have been proposed to build GRN, their convergence and stability have not been investigated enough. In this chapter, we investigate the convergence and stability of building gene regulatory networks (GRN) with dynamic Bayesian networks (DBN). In particular, we address four issues: (1) how DBN can achieve robustness by using priors on the structure and the parameters; (2) how bootstrapping can be used to handle the limited time points and how it can improve the confidences of the networks and the connections; (3) how to validate the networks using synthetic data and protein-protein interaction (PPI) data, and (4) the utility of the methods on real data.

We demonstrate our technique by using synthetic data as well as real data obtained in yeast cell cycle, *M.tuberculosis* and human Hela cell-cycle. Our experiments show that our technique yields regulatory networks that better fit the gene expression data and are biologically more plausible than those obtained with earlier techniques. Furthermore, we look into the key genes and hubs found in the network and their relevance to the biological processes.

We begin this chapter with a description of Bayesian networks (BN) and its extension to dynamic Bayesian networks (DBN) [7], [35]. Suitable priors for learning structure and parameters are investigated. Next, pros and cons of structural learning by using MCMC and GA are compared. Since microarray time-series is expensive, we resort to residual bootstrap to generate additional samples and determine the confidence of estimated edges. Lastly, indices of stability of the model are derived and computed.

2.1 Motivation and Contributions

GRN are causal models that represent regulatory events and indicate how one gene causes another gene to be activated or inhibited. A plethora of compu-

tational approaches have been introduced in the literature to infer GRN from microarray data: for example by Boolean networks [19], [37], linear differential equations [22], concept networks, [23], and Bayesian networks [24], [21], [20], [38]. dynamic Bayesian networks (DBN) have been increasingly used for causal inference of genetic networks [5], [39], [10], [40], [41] because it describes the behaviour of biological GRN very closely.

Microarray data is inherently noisy due to imperfections on hybridizing slides and errors in their fabrication. Gathering gene expressions by using microarrays is very expensive and tedious so they are collected only on a few time points in practice. Because of the limited time points, building gene regulatory networks is an ill-posed computational problem.

Noise and artifacts are inherent in microarray data and their effects on data analysis have to be taken into consideration. Therefore, in order to mitigate the effects of noise and limited number of samples, we investigate priors and propose bootstrapping techniques [9], [42]. Bootstrapping also improves the confidences of networks built and their connections.

Inference using a DBN is NP-hard as the number of graph structures to be evaluated increases exponentially with the number of genes [43]. Heuristic approaches restrict searching a sub-space of models with highest marginal likelihood. The simplest approach called Greedy-Hill climbing accepts a new structure if the probability of a single gene increases and hence does not guarantee the optimal solution [44]. The Expectation-Maximization procedure starts with an initial guess and updates parameters of the ML estimator until convergence. It is useful when there is a lot of data available but can get stuck in local minima [25]. The MCMC simulation optimizes the posterior probability of all the edges jointly in a stochastic manner and hence escapes the local minima. It samples space of possible network structures such that structures with a high likelihood are preferred over structures with a low likelihood at the same time if it jumps into the new structures with a probability. The MCMC simulation has high accuracy though it is computationally very slow and multiple independent runs have to be performed in order to assess the confidence of prediction. Genetic algorithms (GA) searches for a good model on biological evolution by maintaining a population of models which are used to form new generations of models through combining and mutating the current population [45]. A single run of GA is sufficient and parallel processing of structures is a trade-off between speed and accuracy.

Previously, Greedy hill climbing and GA have been compared in prediction of GRN [46]. Consensus modelling of GRN using MCMC and GA have also

been studied [47], [48]. We study the convergence and stability of building GRN from gene expression datasets, by using MCMC simulation. Results are compared with GA. In particular, we are interested in finding the methods that perform best with a small number of time points.

Gene expressions when collected over a sufficiently large number of time points can be used to derive gene regulatory network (GRN) [49]. However, because of the experimental limitations and cost, the number of time points at which gene expressions can be gathered in practice is limited. Hence, the construction of GRN is often subject to inaccuracies and instabilities [2]. The biological noise and variability as well as instrumental artifacts add to the complexity of gene expression data analysis. Sensitivity of DBN was addressed in [39]. However, investigation of their stability in building GRN has evaded the research community. In this chapter, we study the effect of number of time samples on network size.

In the stochastic framework, the ill-posed nature of building GRN is handled by using priors. Parameter priors regularize the model by adding a penalty when the model deviates from a prior structure [50]: such as pseudo counts to account for missing samples. These Dirichlet priors enable modeling of complex interactions among the genes in regulatory networks [24], [51] by providing smoother or less extreme estimates of parameters compared to maximum likelihood. Structural priors reflect our prior knowledge about the average number of parents such as binomial distribution for the mean number of parents [52], [53], [54].

Validation of our methods is done in two ways (1) by using synthetic data, and (2) comparison with known biological knowledge in the corresponding protein interaction data [55]. In order to demonstrate the utility of our methods, we build GRN for these benchmarks gene expression datasets: Yeast cell cycle data, *M.tuberculosis* dna repair data and human Hela cell cycle data. Key genes and hubs and corresponding function are detected from the networks and compared with those reported in literature [56], [57], [34].

2.1.1 Notations used in the Thesis

Let us consider a set $(G = i)_{i=1}^n$ of n genes and their time-series gene expressions. In this thesis, we presume that gene expressions are collected over T equally spaced time points of all genes. Let the dataset of gene expressions be $x = (x_i(t))_{i=1, t=1}^{n, T}$ where $x_i(t)$ represents the gene expression value of i^{th} gene at time point t . In our work, gene expressions are discretized into a set Γ of

d levels: $\Gamma = \{k\}_{k=1}^d$. A level of gene expression indicates the state of a gene. Let a_i be the set of parent genes regulating gene i . Then, the number of states that the set a_i can take is q_i where $q_i = d^{|a_i|}$.

A GRN is a directed graph characterized by its structure s and a set θ of parameters indicating the strength of connections or regulations. The connectivity matrix c is defined as $c = \{c_{ii'}\}_{i=1, i'=1}^{i=n, i'=n}$ where $c_{ii'}(> 0)$ denotes the delay of interactions between genes i and i' .

2.2 Bayesian Networks

Bayesian networks (BN) provide a stochastic framework for building gene regulatory networks [58]. Nodes of BN represent the genes and connections represent the interactions between genes of GRN. Strengths of the connections between genes are represented by conditional probabilities of genes regulated by their parent genes.

Figure 2.1: Illustration of a gene regulatory network represented in a Bayesian network. Nodes represent genes and connections represent causal interactions: x_i represents the activity of gene i and $p(x_i|x_j)$ represents the regulation or the conditional probability of gene i having state x_i when gene j has the state x_j .

A BN is a graphical model that represents a joint multivariate probability distribution of random variables by capturing the properties of conditional independencies among the variables. It is a directed acyclic graph (DAG) having a structure and a set of parameters that represent the strengths of connections

by conditional probabilities. Fig. 2.1 illustrates a BN with five nodes (or genes). The BN decomposes the likelihood of gene expressions into a product of conditional probabilities by assuming independence of non-descendant genes, given their parents :

$$p(x|s, \theta) = \prod_{i=1}^n p(x_i|a_i, \theta_i) \quad (2.1)$$

where $x = (x_1, x_2, \dots, x_n)$ denotes the vector of gene expressions of n genes, $p(x_i|a_i, \theta_i)$ denotes the conditional probability of gene expression x_i given its parent gene expressions a_i , and θ_i denotes the parameters of the conditional probabilities .

2.3 Dynamic Bayesian Networks

The acyclic condition of BN does not allow self- and feedback- regulations of genes, which are essential characteristics of GRN [49], [59]. Therefore, dynamic Bayesian networks (DBN) have recently become popular in building GRN from gene expression data mainly due to their ability to model causal interactions as well as feedback regulations [35]. The DBN models regulatory networks by using information from one time point to the next and represents temporal expression patterns by unrolling the network over time.

The DBN makes the following assumptions [60], [61]:

- (1) The genetic regulation is a first-order Markovian process. That is the expression state of one gene at one time point is dependent only on the expression state of other genes observed at the previous time point.
- (2) The causal relationships between genes are invariant over time. That is, the set of variables and probability definitions of the DBN remain same (stationary) throughout the time series.

Because of the above assumptions, the GRN built using DBN are synchronous and stationary. These assumptions are not typical of biological networks. Therefore models that could handle variable delays are necessary. The DBN structure is obtained by unrolling the transition network over time (Fig. 2.2). By assuming that the network of interactions between time instances t and $t + 1$ are independent of time, the likelihood of time-series is given by

$$P(x) = \prod_{i=1}^n p(x_i(t)) = \prod_{i=1}^n \prod_{t=1}^{T-1} p(x_i(t+1)|x_i(t), \theta_i) \quad (2.2)$$

Figure 2.2: Transition network of a gene regulatory network of n genes, represented by a dynamic Bayesian network (DBN)

A first-order DBN is defined by a transition network of interactions between a pair of structures connecting genes at time instances t and $t + 1$. In time instance $t + 1$, the parents of genes are those specified in the time instant t (see Fig. 2.2). Hence, similar to an autoregressive model the probability of a gene is dependent on parents in time t as well as $t + 1$ as in a first-order Markov model given by (2.3)

$$p(x|s, \theta) = \prod_{i=1}^n p(x_i^t | a_i^t, \theta_i^t) \prod_{i=1}^n p(x_i^{t+1} | a_i^{t+1}, \theta_i^{t+1}) \quad (2.3)$$

where $x = (x_1, x_2, \dots, x_n)$ denotes the vector of gene expressions of n genes, $p(x_i^t | a_i^{t-1}, \theta_i^{t-1, t})$ denotes additional parameters for the conditional probability of gene expression x_i given its parent gene expressions a_i from previous time point, and $\theta_i^{t-1, t}$ denotes the parameters of the conditional probabilities of the transition network. While edges in each network have to be acyclic, the transition network can have any edges and hence can model loops as well as time delays. Implementation simply adds additional node in the BN to model the delay resulting in a network of $2 \times n$ nodes, and input data is shifted by a single time point to learn these duplicate nodes.

The optimal structure s^* is obtained by maximizing the posterior probability of s given data x . From Bayes theorem, the optimal structure s^* is given by

$$s^* = \max_s p(s|x) = \max_s p(s)p(x|s) \quad (2.4)$$

where $p(s)$ is the probability of the network structure and $p(x|s)$ is the likelihood of expression data given the network structure.

Given the set of conditional distributions with parameters $\theta = \{\theta_i\}_{i=1}^n$, the likelihood of data is given by

$$p(x|s) = \int p(x|s, \theta) p(\theta|s) d\theta \quad (2.5)$$

To find the likelihood in (2.5) in order to obtain the optimal structure as in (2.4), the data is pre-assumed to be either Gaussian or multinomial [62]. Though the Gaussian model makes determination of the structure computationally tractable, it is unable to capture complex regulatory interactions present among genes in biological pathways [24]. The multinomial assumption requires quantization of data. Proper, quantization of data also stabilizes the noise in the measurements at a loss of important information. The multinomial assumption is often preferred in building gene regulatory networks. Throughout this thesis, we assume that gene expressions are distributed multinomially because it is closer to biological regulatory networks [50]. Multinomial assumption allows modelling complex interactions like those observed in biological networks.

In what follows gene expressions are assumed to be discretized into a set of d levels: $\{k\}_{k=1}^d$. A particular level of gene expression corresponds to a state of the gene. Let the number of states that the set a_i of parents regulating the gene i take to be q_i and $\theta_{ijk} = p(x_i = k | a_i = j)$ where N_{ijk} be the number of instances of θ_{ijk} that occur in the training data. Using the property of decomposability [60], (2.1) can be written as:

$$p(x|s, \theta) = \prod_{i=1}^n \prod_{j=1}^{q_i} \prod_{k=1}^d \theta_{ijk}^{N_{ijk}} \quad (2.6)$$

The above likelihood represents the likelihood of d -dimensional time-series gene expression data, having conditional probability values $(\theta_{ijk})_{i=1, j=1, k=1}^{n, q_i, d}$. Note that only the counts $N_{ijk} = \sum_{t=1}^T \delta(x_i = k | a_i = j)$ are necessary to determine the likelihood. These counts provide "sufficient statistics" to represent multinomial gene expression time-series.

2.4 Parameter Learning

2.4.1 Maximum Likelihood Estimates of Parameters

Taking the logarithm of (2.6), the log-likelihood is given by

$$\log p(x|s, \theta) = \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^d N_{ijk} \log \theta_{ijk} \quad (2.7)$$

in which the parameters are constrained by

$$\sum_{k=1}^d \theta_{ijk} = 1 \forall i, j \in a_i \quad (2.8)$$

By combining (2.7) with constraints in (2.8) the Lagrange equation is obtained:

$$J = \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^d N_{ijk} \log \theta_{ijk} + \sum_{i=1}^n \sum_{j=1}^{q_i} \lambda_{ij} \left(\sum_{k=1}^d \theta_{ijk} - 1 \right) \quad (2.9)$$

Taking the derivative with respect to θ_{ijk} , it can be shown that (Appendix A.2) the maximum likelihood (ML) estimate of parameters is given by

$$\hat{\theta}_{ijk} = \frac{N_{ijk}}{\sum_{k=1}^d N_{ijk}} \quad (2.10)$$

And by substituting $\hat{\theta}_{ijk}$ in (2.7), the log-likelihood of the data is given by :

$$\log p(x|s) = \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^d N_{ijk} \log \frac{N_{ijk}}{\sum_{k=1}^d N_{ijk}} \quad (2.11)$$

The ML estimates of the parameters are known to be good when a large number of data samples are available. That is, when the time-series are long enough or the number of counts are large enough. However, if the dataset is small, the estimates are sensitive to noise and lead to over-fitting of the data. One way to circumvent this over-fitting of data is to use priors.

2.4.2 Prior of Parameters

By introducing priors, the robustness of the parameters and over-fitting of the data can be improved. The parameters of individual conditional probabilities are assumed locally independent. That is, the parameters associated with each state of the parents of a gene are independent. Therefore,

$$p(\theta|s) = \prod_{i=1}^n p(\theta_i|s) = \prod_{i=1}^n \prod_{j=1}^{q_i} p(\theta_{ij}|s) = \prod_{i=1}^n \prod_{j=1}^{q_i} \prod_{k=1}^d p(\theta_{ijk}) \quad (2.12)$$

Let us assume that the prior distribution of θ_{ijk} is given by Dirichlet distribution with hyper-parameters α_{ijk}

$$p(\theta|s) = Z(\alpha_{ijk}) \prod_{i=1}^n \prod_{j=1}^{q_i} \prod_{k=1}^d \theta_{ijk}^{\alpha_{ijk}-1} \text{ where } Z(\alpha_{ijk}) = \frac{\Gamma(\sum_k \alpha_{ijk})}{\prod_k \Gamma(\alpha_{ijk})} \quad (2.13)$$

Dirichlet distribution are flexible with a free parameter for expected number of samples, hence they not only gives more biologically plausible GRN but also provides a mathematical tractable framework [51]. Then the denominator is the posterior distribution given the data (integrated over the parameter space). By substituting (2.13) in (2.6), the posterior distribution is given by.

$$p(\theta|s, x) \propto p(x|s, \theta)p(\theta|s) \propto \prod_{i=1}^n \prod_{j=1}^{q_i} \prod_{k=1}^d \theta_{ijk}^{\alpha_{ijk} + N_{ijk} - 1} \quad (2.14)$$

The normalization term of the posterior is updated with sufficient statistics N_{ijk} from time-series x , giving

$$p(\theta|s, x) \propto p(x|s, \theta)p(\theta|s) = Z(\alpha_{ijk} + N_{ijk}) \prod_{i=1}^n \prod_{j=1}^{q_i} \prod_{k=1}^d \theta_{ijk}^{\alpha_{ijk} + N_{ijk} - 1} \quad (2.15)$$

By substituting (2.6), (2.13) and (2.15) in (2.12), we are only left with the ratio of the two normalizing constants :

$$p(x|s) = \frac{Z(\alpha_{ijk})}{Z(\alpha_{ijk} + N_{ijk})} = \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^d \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \quad (2.16)$$

where $N_{ij} = \sum_{k=1}^d N_{ijk}$, and Γ is the gamma function. The above likelihood (3.8) is also known as the Bayes Dirichlet metric (BDe) [35].

Taking the derivative with respect to θ_{ijk} , the maximum a posterior(MAP) of parameters is given by (Appendix A.3):

$$\hat{\theta}_{ijk} = \frac{N_{ijk} + \alpha_{ijk} - 1}{\sum_{k=1}^d N_{ijk} + \sum_{k=1}^d \alpha_{ijk} - d} \quad (2.17)$$

The Dirichlet prior performs well even when the data size is small. Though consideration should be given as it over-penalizes for larger datasets [51].

2.5 Structure Learning

The structure of GRN is obtained by the maximum a posteriori (MAP) estimation:

$$s^* = \arg \max_s p(x|s, \hat{\theta})p(s) \quad (2.18)$$

where $p(s)$ is the prior.

The prior $p(s)$ over the network plays a relatively minor role, as the log of ML grows linearly with number of sampled structures while the prior remains constant. Different approaches for network prior include informative Gibbs

prior, minimum description length of graph, geometric distribution on number of parents, and also ordering of nodes. The structural prior seems to serve as a penalty for overly complex models. The uniform prior assumes that all structures are equally likely. It is desirable to have small structural variance and avoid topologies with either very large or very small number of parents.

A prior distribution is based on tractability, computational convenience and desirable frequency of missing samples, while key parameters of a prior can be chosen, a good prior is spread over all parameter space. Data sampled from a geometric distribution can be easily modeled as a Dirichlet process and hence is commonly used to prior for topology. Hence, we resort to a geometric distribution on the topology prior of each gene. The prior probability of a gene i , given its parent set a_i is given by

$$p(s_i) = q^{|a_i|}(1 - q)^{n-|a_i|} \text{ and } p(s) = \frac{1}{Z} \prod_{i=1}^n p(s_i) \quad (2.19)$$

where $Z = \sum_{s \in S} \prod_{i=1}^{n_s} p(s_i)$ is a normalization constant and S is the set of all possible graphical models. Using the modularity of DBN the normalization constant Z sums over all possible parent configurations for each node and is given by

$$Z = \prod_{i=1}^n \sum_{i=1}^n \binom{n}{i} q^i (1 - q)^{n-i} \quad (2.20)$$

where $\binom{n}{i}$ is the number of possible ordering of i parents [52]. That is, the number of parents $|a_i|$ follows a binomial distribution where the mean number of parents $|\hat{a}_i| = n \times q$. Our choice of q reflects our prior knowledge about the average number of parents [53].

The modified cost function in (2.15) is given by

$$p(s|x) = \frac{1}{Z} \prod_{i=1}^n p(s_i) \prod_{j=1}^{J_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^K \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \quad (2.21)$$

To find the optimal structure by applying the above cost function we use MCMC simulations or genetic algorithms. Initialization of the structure is necessary for the above optimization procedure. In the case of MCMC a burn-in period of sampling can help minimize the effect of a starting structure.

2.5.1 Initialization of the structure

The starting structure is known to significantly affect convergence of the optimization methods used in deriving the optimal network structure. Here the connectivity $c_{ii'}$ between the expressions of genes i and i' is initialized by using mutual information(MI) between the two genes [63]. It is presumed that mutual information between the genes is maximum. The MI between gene i and i' is given by

$$MI(i, i') = H(i) + H(i') - H(i, i') \quad (2.22)$$

where the entropies H are given by

$$\begin{aligned} H(i) &= - \sum_t \sum_{x_i(t) \in \Delta} p(x_i(t)) \log p(x_i(t)) \\ H(i, i') &= - \sum_t \sum_{(x_i(t), x_{i'}(t)) \in \Delta^2} p(x_i(t-1), x_{i'}(t)) \log p(x_i(t-1), x_{i'}(t)) \end{aligned} \quad (2.23)$$

If the in-degree of a gene is restricted, then connections are initialized as follows [10]:

$$c_{ii'} = \begin{cases} 1 & \text{if } MI(ii') > \alpha \\ 0 & \text{if } M(i, i') < \alpha \text{ or } |a_i| > n/3 \end{cases} \quad (2.24)$$

where $|a_i|$ is number of parents of gene i and α is the threshold for mutual information. We randomize the order of genes during initialization to avoid duplicate starting individuals .

2.5.2 MCMC Simulations

In order to find the optimal structure, a Markov chain of structures is formed with Markov chain Monte Carlo (MCMC) simulation which converges to the optimal structure at the equilibrium. The Metropolis-Hastings method of MCMC is adopted, which associates an acceptance mechanism when new sample structures are drawn (Algorithm 1). The acceptance of new structure s^{new} is given by

$$\min \left\{ 1, \frac{p(s^{\text{new}})}{p(s)} \cdot \frac{p(s^{\text{new}}|x)}{p(s|x)} \cdot \frac{q(s^{\text{new}}|s)}{q(s|s^{\text{new}})} \right\} \quad (2.25)$$

where the Metropolis-Hastings acceptance ratio:

$$\frac{q}{1-q} \cdot \frac{p(s^{\text{new}}|x)}{p(s|x)} \cdot \frac{q(s^{\text{new}}|s)}{q(s|s^{\text{new}})} \quad (2.26)$$

Algorithm 1 Markov Chain Monte Carlo

```

Initialize topology of  $s = s_0$ 
repeat
  Generate  $s^{new}$  by elementary operations
  Find  $N^{new}$  and  $N^{old}$  corresponding to  $s^{new}$  and  $s$ 
  Find acceptance ratio  $\alpha$  given by (2.26)
  if  $\alpha \geq 1$  then
     $s = s^{new}$ 
  else
    if  $rand[0, 1] \geq \alpha$  then
       $s = s^{new}$ 
    end if
  end if
until convergence

```

when adding an edge and the prior ratio is inverted when deleting an edge.

Sampling new structures by using the above procedure generates a Markov chain converging in distribution to the approximate posterior distribution. By taking average over sampled structures after a burn-in period we can compute the integral over parameters in (2.5). In practice, a new network structure is proposed by applying one of the elementary operations such as deleting, reversing, or adding an edge, and then discarding those structures that violate the acyclic condition. The first and second term of the acceptance ratio, the ratio of likelihoods, is computed using (2.21). The third term or Hastings ratio is obtained by

$$\frac{q(s^{new}|s)}{q(s|s^{new})} = \frac{N^{new}}{N} \quad (2.27)$$

where N denotes the size of the neighborhood obtained by elementary operations on structure s and counting of the valid structures. The network is initialized by using mutual information [64], [65].

2.5.3 Genetic Algorithm

Genetic algorithms (GA) are a set of evolutionary algorithms that are used to solve a wide variety of optimization problems. Compared to MCMC, GA converges faster but does not guarantee optimal solutions. In this section, we present a genetic algorithm that finds the optimal structure of GRN [45].

The cost function (or the fitness function) of the GA is the maximum likelihood such as BDe score in (2.21). The GA performs the optimization by searching through a population of structures or connectivity matrices by iteratively updating members of the solution population by using Algorithm 2.

After starting with a population of random structures, crossover and mutations operations are performed until the optimal structure is evolved. The crossover involves swapping several rows of the connectivity matrix of two parents, resulting in structures with higher fitness. The mutations alter a single cell or element in the connectivity matrix.

Here we run the GA for Q generations or until the change of the fitness score becomes less than a particular value. As low lying structures can easily dominate the others leading to premature search convergence, a minimum similarity threshold is maintained in each generation. The parameters of GA were found heuristically for maximum likelihood of the structure. Similar to MCMC, the generation of structures is initialized using MI. We randomize the order of genes during initialization for each individual to enable multiple start points.

Algorithm 2 Genetic Algorithm for Structure Learning

Initialize N random individuals with connectivity matrix $\{c_{i,i'}\}_{i=1,i'=1}^{n,n}$ by using MI
Iterate for Q generations until convergence
repeat
 Select elite individuals the with highest fitness: $score_q$ (2.21)
 repeat
 Select two individuals based on the fitness by using Roulette wheel
 Crossover two individuals by swapping last few rows
 Accept the new individuals if population similarity $> th_s$
 $j = j + 2$
 until $j < N$
 Mutation: Select a random individual and invert a random cell in it
until $q < Q$ or convergence
Build the gene regulation matrix from an elite individual

2.6 Stability

The stability of building GRN was evaluated by drawing synthetic time-series gene expressions from a network of known topology. The structure s is defined by a connectivity matrix $c = \{c_{ii'}\}_{n \times n}$ where $c_{ii'} = 0$ or $\neq 0$ depending on the presence or absence of a regulatory connection between two genes, i and i' . Stability was evaluated by generating samples of gene expression time-series from GRN, represented by a DBN with a known structure and parameters. The distance based similarity criteria were used to evaluate the stability of estimating structure and parameters of GRN [39].

2.6.1 Simulation of Gene Expression Time-Series

The likelihood of gene expressions $x(t)$ of genes at t is given by

$$p(x(t)|s, \theta) = \prod_{i=1}^I p(x_i(t)|x(t-1), \theta_i) \quad (2.28)$$

and the gene expression value of gene i is estimated by

$$\hat{x}_i(t) = \arg \max_k p(x_i(t) = k | \theta_{ijk}, x(t-1) = j) \quad (2.29)$$

If the gene expressions at $t = 1$ is initialized, $\{x_i(1)\}_{i=1}^I$, subsequent time points are generated by sampling from GRN by using (2.47) and adding Gaussian noise $N(0, \sigma^2)$. Algorithm 3 lists the pseudo code for generating synthetic time series data.

Algorithm 3 Generating samples of gene expression data

Given: structure s , parameters θ , and Gaussian noise s.d. σ

Let $x(1) = (x_i(1) = \text{rand}[1, K])_{i=1}^I$

for $t = 2$ to T **do**

for $i = 1$ to I **do**

if $x(t-1) = j$ and $j \in J_i$ **then**

$\hat{x}_i(t) = \arg \max_k p(x_i(t) = k | \theta_{ijk}, x(t-1) = j)$

else

$\hat{x}_i(t) = 0$

end if

$\epsilon_i \sim N(0, \sigma^2)$

$x_i(t) = \hat{x}_i(t) + \epsilon_i$

end for

end for

Return: dataset $(x(t))_{t=1}^T$

2.6.2 Stability of Structure Estimation

We propose a similarity based stability criterion for evaluating GRN building algorithms, which is measured by the mean over all pairwise Hamming distances among all the detected connections over different datasets.

Let $\{x^b\}_{b=1}^B$ be a set of B samples of gene expression data and s^b be the GRN derived from b -th dataset x^b . Consider two GRNs with connectivity matrices $c^b = \{c_{i,i'}^b\}_{n \times n}$ and $c^{b'}$ derived from datasets x^b and $x^{b'}$, respectively. The similarity of two networks is obtained by the average Hamming distances over all regulatory connections:

$$\rho(c^b, c^{b'}) = 1 - \frac{1}{N^b + N^{b'}} \sum_{i=1}^I \sum_{i'=1}^I d(c_{i,i'}^b, c_{i,i'}^{b'}). \quad (2.30)$$

where d denotes the Hamming distance between the connections, N^b denotes the number of connections in network and s^b have connectivity matrix c^b . The Hamming distance takes consider both the presence (1) and the absence (0) of regulatory connections between two networks. The stability of the network is in the range of $[0, 1]$ where a higher value denotes a higher stability of GRN inference algorithm.

Using the stability of two networks in (2.30), the stability of the GRN structure building algorithm is obtained by averaging over B number of structures. By averaging pairwise stability given in (2.30), the stability of the network structure is obtained as:

$$\rho_{\text{structure}} = \frac{2}{B(B-1)} \sum_{b=1}^B \sum_{b'=b+1}^B \rho(c^b, c^{b'}). \quad (2.31)$$

The stability of an independent connection is defined as the average of bootstrap samples

$$\rho_{\text{connection}}(i, i') = \frac{\sum_{b=1}^B c_{i,i'}^b}{B} \quad (2.32)$$

2.7 Existing Techniques

There are a plethora of techniques in the literature for building GRN. Our methods were compared with two state of art GRN building algorithms : multivariate autoregression (MVAR) models and Gaussian Bayesian networks (GBN), because of their wide use and close similarity to DBN.

2.7.1 Multivariate Autoregressive Models

An r -order multivariate autoregressive (MVAR) model represents linear GRN with delays:

$$x(t) = \sum_{\tau=1}^r \beta^\tau x(t - \tau) + \varepsilon(t) \quad (2.33)$$

where $\beta^\tau = \{\beta_{i,j}^\tau\}_{n \times n}$ denotes the matrix of regression coefficients corresponding to delay of τ , and $\varepsilon(t) = (\varepsilon_i(t))_{i=1}^n$ denotes i.i.d. zero mean additive Gaussian noise. The regression coefficients $\beta_{i,j}^\tau$ when $i \neq j$, represents the interaction between the two genes: i and j [66]. The structure of GRN is obtained by forcing small coefficients to zero by penalizing for the magnitude of coefficients.

2.7.1.1 Estimating Regression Coefficients

Suppose j^{th} sample of time-series is denoted by the row $y^j = [x_1(j), x_2(j), \dots, x_n(j)]$ and let $x^j = [y^{j-1}, y^{j-1}, \dots, y^{j-r}]$ denote a row of rn elements, $\beta = [\beta^1, \beta^2, \dots, \beta^n]^T$ denote a matrix of size $rn \times n$, and $\varepsilon^j = [\varepsilon_1(j), \varepsilon_2(j), \dots, \varepsilon_n(j)]$ denote the corresponding innovations, then the multivariate model can be written in the standard multivariate autoregressive (MAR) form [67] :

$$y^j = x^j \beta + \varepsilon^j \quad (2.34)$$

If the j th row of Y , X , and E , are y^j , x^j , and e^j , respectively, and there are J samples, we can write the MVAR using matrices:

$$Y = X\beta + E \quad (2.35)$$

where Y is a $J \times n$ matrix, X is a $J \times rn$ matrix, β is a $rn \times n$ matrix, and E is a $J \times n$ matrix. The MVAR coefficients are estimated using least squares:

$$\hat{\beta} = (X^T X)^{-1} X^T Y \quad (2.36)$$

and the maximum likelihood estimates of the covariance of innovations are estimated by

$$S = \frac{1}{J - n^2 r} (Y - X\hat{\beta})^T (Y - X\hat{\beta}) \quad (2.37)$$

Lastly, the ML parameter covariance matrix $\hat{\Sigma}_b$ is given by :

$$\hat{\Sigma}_b = S \otimes (X^T X)^{-1} \quad (2.38)$$

where $\otimes$ denotes the Kronecker product.

2.7.1.2 Testing the Significances of Edges

For order r , we obtain $k = (n \times n) \times r$ MAR coefficients. This essentially gives connectivity between two genes delays up-to r . Moreover, rarely a MAR coefficient is exactly zero. Hence it is vital to determine statistically significant non-zero weights, i.e., the presence of edges.

A statistical method to identify significant edges was developed by [67]. In this method, for gene i and j , a vector of coefficient γ is first defined where each element of γ denotes a value of MVAR coefficient at different time order. To estimate the weights of connections between two genes, a sparse matrix C of size $k \times k$ is defined such that

$$\gamma = C^T \beta \quad (2.39)$$

γ is assumed to be Gaussianly distributed : $p(\gamma) = N(\mu_\gamma, \Sigma_\gamma)$, where mean (μ_γ) and covariance (Σ_γ) are given by

$$\mu_\gamma = C^T \hat{\beta}_b \quad (2.40)$$

$$\Sigma_\gamma = C^T \hat{\Sigma}_b C \quad (2.41)$$

Here, $\hat{\beta}_b = \text{vec}(\hat{\beta}_b^*)$ and $\hat{\Sigma}_b$ are Bayesian estimates of the parameters of the posterior distribution of regression coefficients, which can be obtained using algorithm given by [67].

The test statistic is then formed as

$$d = T \mu_\gamma^T \Sigma_\gamma^{-1} \mu_\gamma \quad (2.42)$$

2.42 is sum of $q = \text{rank}(\Sigma_\gamma)$ independent, squared Gaussian variables. It has a χ^2 distribution $p(d) = \chi^2(q)$.

2.7.2 Gaussian Bayesian Networks (GBN)

Gaussian BN assume that the gene are multivariate Gaussian. That is, gene expression can be described with mean μ and covariance matrix Σ of size $n \times n$ [68]. The joint probability of the network can be the product of a set of CPD's given by

$$p(x_i | a_i) \sim \mathcal{N} \left(\mu_i + \sum_{j \in a_i} (x_j - \mu_j) \beta, \Sigma'_i \right) \quad (2.43)$$

where $\Sigma'_i = \Sigma_i - \Sigma_{ia_i} \Sigma_{a_i}^{-1} \Sigma_{ia_i}^T$

and β denotes the regression coefficient matrix and Σ'_i is the conditional variance of x_i given its parents set a_i where Σ_{ia_i} is the covariance between time-series x_i and variables in a_i and Σ_{a_i} is the covariance matrix of a_i .

2.8 Bootstrapping

The idea behind bootstrapping is a simulation of a statistic or statistical procedure from a distribution that is estimated from observed data [69]. Using the statistic or statistical distribution(s) derieved from the available samples and new estimates of parameters and structure are found. Because of the limited

number of time samples and in order to evaluate the confidence of the networks and connections derived, we resort to bootstrapping. First, we indicate how bootstrapping can be used to enhance gene expression analysis by generating additional samples. Second, we use bootstrapping to evaluate the confidences of the networks generated and the connections within.

The empirical process in the multivariate autoregressive model converges weakly to a certain Gaussian process. It has been shown that bootstrapped empirical process converges weakly to the same Gaussian process [70].

2.8.1 Vector Sieve Bootstrap

Sieve bootstrapping relies on sieve approximation as data generating process by a family of parametric models [42].

Vector sieve bootstrap generates samples from an empirical distribution derived from a distribution of innovations. The cumulative distribution of the residuals or innovations $\varepsilon_i(t)$, say $F_{\varepsilon_i}(\cdot)$, can be evaluated as

$$\hat{F}_{\varepsilon_i}(h) = g_n(\varepsilon_i \leq h) = \frac{1}{(T-r)} \sum_{t=r+1}^T \delta(\varepsilon_i(t) - \bar{\varepsilon}_i \leq h)$$

where

$$\varepsilon_i(t) = x_i(t) - \sum_{\tau=1}^r \beta_{i,\tau}^\tau x(t-\tau) \quad (2.44)$$

where $\bar{\varepsilon}_i = \frac{1}{T-r} \sum_{t=r+1}^T \varepsilon_i(t)$, $\beta_{i,\cdot}^\tau = (\beta_{i,1}^\tau, \beta_{i,2}^\tau, \dots, \beta_{i,n}^\tau)$, and $g_n(\varepsilon_i)$ denotes the distribution of innovations $\varepsilon_i(t)$. The joint density of innovations $g_n(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n) = \prod_{i=1}^n g_n(\varepsilon_i)$.

Let $\beta = [\beta^1, \beta^2, \dots, \beta^r]^\top$ denote the matrix by stacking all β^τ sequentially down from $\tau = 1$ at the top. And $\hat{\beta} = \text{vec}(\hat{\beta})$ where $\text{vec}(\cdot)$ denotes the columns of $\hat{\beta}$ being stacked on top of each other. That is, $\hat{\beta}$ represents the vector of all regression coefficients. Using least squares, the parameter set can be estimated. The estimated parameter set $\phi = \{\hat{\mu}, \hat{\beta}, \hat{g}_n(\varepsilon)\}$ characterizes the empirical distribution $\hat{f}_n$. The bootstrap time samples $x^*(t)$ is given by

$$x^*(t) - \hat{\mu} = \sum_{\tau=1}^r \hat{\beta}^\tau (x^*(t-\tau) - \hat{\mu}) + \varepsilon^*(t) \quad (2.45)$$

with $\varepsilon^*(t)$ is an i.i.d. innovations drawn from $\hat{g}_n$. Thereby, the VAR-sieve bootstrap time series $x^* = (x^*(1), x^*(2), \dots, x^*(T))$ are obtained from the empirical distribution $\hat{f}_n$. The sampling in practice begins with $x(t) = \hat{\mu}$ for all $t = 1, \dots, r$, and then simulate $x^*(t)$ for $t = r+1, \dots, n$ according to (2.45).

2.8.2 Probabilistic Residual Bootstrap

In this section, we introduce a novel probabilistic residual bootstrap technique that can be used to generate samples from GRN built with DBN. Since, gene expression time-series are short-time series, only a few block samples are available. Similar to sieve bootstrap we can resample residuals from this error distribution of MAP structure s and corresponding CPD θ . These residuals can be used to generate additional bootstrap samples.

The likelihood of gene expressions $x(t)$ of genes at t is given by

$$p(x(t)|s, \theta) = \prod_{i=1}^I p(x_i(t)|x(t-1), \theta_i) \quad (2.46)$$

and the gene expression value of gene i is estimated by

$$\hat{x}_i(t) = \arg \max_k p(x_i(t) = k | \theta_{ijk}, x(t-1) = j) \quad (2.47)$$

If the gene expressions at $t = 1$ is initialized the gene expressions, $\{x_i(1)\}_{i=1}^I$ at subsequent time points are generated by sampling from GRN by using (2.47).

If ML parameters $\{\theta_{ijk}\}_{i=1, j=1, k=1}^{n, q_i, d}$ are first specified, it is possible to determine the corresponding error probabilities $\{\epsilon_{ij}\}_{i=1, j=1}^{n, q_i}$ from the data [71]. By assuming that expressions of all genes are quantized into an equal number of levels the counts of residual gene expression are obtained as

$$\epsilon_{i, \hat{x}_i(t)} = \delta(\hat{x}_i(t) \neq \arg \max_k \theta_{ijk} | a_i(t-1) = j) \quad (2.48)$$

Assuming that the errors are time-independent, the error counts M_{ijk} and corresponding error probability can be computed as

$$\epsilon_{ij} = \frac{M_{ijk}}{T} \text{ where } M_{ijk} = \sum_{t=1}^T \epsilon_{ijk} \forall k \quad (2.49)$$

For a given short time-series we predict s and θ using BDe metric of DBN. The structure s is defined by a connectivity matrix $c = \{c_{ii'}\}_{n \times n}$ where $c_{ii'} = 0$ or $\neq 0$ depending on the presence or absence of a regulatory connection between two genes, i and i' . Next, B bootstrap samples are generated using θ and random errors are introduced from distributions ϵ_{ij} according to Algorithm (4).

In order to define a $1-p$ confidence interval, where p is the confidence level, say b bootstrap samples of data are drawn: $x^1*, x^2*, \dots, x^b*$. Then, the lower and upper half intervals can be approximated by $\underline{\delta} = p(x|s) - p(x|s_{0.05})$ and $\bar{\delta} = p(x|s_{0.95}) - p(x|s)$, where $p(x|s)$ is the average likelihood of the structures. The confidence interval then becomes $[p(x|s) - \bar{\delta}, p(x|s) - \underline{\delta}]$. Model validation can be done through detection in changes of parameters.

Algorithm 4 Bootstrapping time series

Given: data $x(t)$, structure s , CPD θ and error distributions ϵ :

```

for  $t = 1$  to  $T$  do
  for  $i = 1$  to  $n$  do
     $a_i(t-1) = (x_{i'}(t-1)c_{ii'})_{i'=1}^n$ 
     $\epsilon_i(t) = \text{sample}\{\epsilon_{ij} | j = a_i(t-1)\}$ 
     $x_i(t) = \arg \max_k \theta_{ijk}$ 
    if  $\text{rand}[0, 1] \leq \epsilon_i(t)$  then
       $x_i(t) = (\text{rand}\{k\}_{k=1}^d)_{i=1}^n$ 
    end if
  end for
end for

```

2.8.3 Validation

Gene regulation is highly complex process. For example time-delays are usually of very high order and a model approximates them using lower-order delays [72]. Since, genes are involved in multiple interleaving activities in the cell we looked at higher-order features whiles validating our network. Earlier tdGRN [73] tried to validate delays by partitioning the genes into parent set (the regulators) and child set (the target). The method then associates delays between the two sets. We show validation of our results by employing protein-protein interaction (PPI) data. In order to validate higher delay interactions we look at the cascades of genes in the GRN, which corresponds to interactions in PPI network [36].

2.8.3.1 k -skip Validation

In addition to the interaction features $f(y_t, y_{t-1})$ present in the network S where y_{t-1} is gene label of parent of y_t . We use additional skip-edges to represent dependence between distant genes. For example, a second-order Markov model for validation has auxiliary feature label z defined by state triples $z_t = (y_{t-2:t})$. The new label z has a second-order Markov dependency to the labels y predicted by our model [13]. We can hence generalise these new interaction features as $h(y_{s_t}, y_t)$ where $s_t - t$ is the order of the skip feature of arbitrary start and end nodes in the network. Since reduced complexity models are used to capture the most important dynamics. Such features can accommodate for certain mismatches in predicted interactions.

Our validation dataset in the protein-protein interaction network (PPIN). We validate against the expanded feature set using k -order validation for each interaction available in PPIN. For two genes g_i and g_j , k -order validation

looks for features $h(y_{t-k}, y_t)$ and corresponding sequence $z_t = y_{t-k:t}$. This can be done by checking for a path of k genes $(g_1, g_2, \dots, g_k)$ such that any gene in the cascade interacts with the next gene.

The simplest is called the first-order Markov model where one gene is parent of second gene. These models are defined as follows:

First-order Model: Validates PPI interaction (g_i, g_j) in the predicted network using the interaction feature $f(y_{t-1}, y_t)$. Hence protein g_i corresponds to label y_{t-1} and protein g_j corresponds label y_t

Second-order Model: Validates PPI interaction (g_i, g_j) in the predicted network using the interaction feature $f(y_{t-1}, y_t)$ as well as second-order features $h(y_{t-2}, y_t)$ using auxiliary labels $z_t = y_{t-2:t}$. Hence protein g_i corresponds to label y_{t-2} and protein g_j corresponds label y_t

Third-order Model: Validates PPI interaction (g_i, g_j) in the predicted network using the interaction feature $f(y_{t-1}, y_t)$, second-order skip features as well as third-order features $h(y_{t-3}, y_t)$ using auxiliary labels $z_t = y_{t-3:t}$. Hence protein g_i corresponds to label y_{t-3} and protein g_j corresponds label y_t

Fourth-order Model: Validates PPI interaction (g_i, g_j) in the predicted network using the interaction feature $f(y_{t-1}, y_t)$, second-order and third-order skip features, as well as fourth-order features $h(y_{t-4}, y_t)$ using auxiliary labels $z_t = y_{t-4:t}$. Hence protein g_i corresponds to label y_{t-4} and protein g_j corresponds label y_t

The Algorithm 5 describes how k -order features are identified in predicted model. We define a boolean function $getintk(g_i, g_j, k)$ which returns 1 if a k -order feature exists between g_i and g_j and 0 otherwise. The function does so by recursively looking for $k - 1$ order features so it can efficiently span the GRN.

We start by defining a set A_i of children genes of gene g_i in the predicted GRN. If g_j belongs to this set then a first-order interaction is reported. Otherwise we call algorithm again with $k = 2$, here for each child m in set A_i we look for a first-order interaction between $A_{i,m}$ and g_j . The algorithm is called k times for $k = (1, 2, \dots, r)$ so as to find the lowest order interaction feature corresponding to a known PPIN. Algorithm 5 lists the pseudo code that can perform the validation recursively.

2.9 Experiments on Synthetic Data

In this section, experiments are performed to investigate convergence and stability of building GRN by using DBN. The results are compared with GBN and

Algorithm 5 k -skip Validation

```

 $A_i$  is array of children of gene  $g_i$ 
 $getintk(g_i, g_j, k)$ 
 $r = 0$ 
if( $k == 1$ )
if  $g_j \in A_i$  then
   $r = 1$ 
else
   $m = 0$ 
  repeat
     $r = getintk(A_i[m], g_i, k - 1)$ 
    if  $r == 1$  then
      break
    end if
     $m \leftarrow m + 1$ 
  until  $m == len(A_i)$ 
end if

```

MVAR techniques. Also, experiments are performed to test the bootstrapping technique, effectiveness of the probabilistic residual and the use of prior.

In order to quantitatively evaluate the stability and convergence of building GRN, multiple time-series datasets of gene expressions were generated from scale-free networks with varying number of genes (nodes), using Barabasi-Albert model [74]. The topology of biological networks are very much close to scale-free networks. The structures of simulated GRN were first built by initiating the networks with a small number of nodes. New nodes and edges were added with preferential attachment because the probability of addition of new nodes to the existing node is not uniform. A node with a high number of edges attracts a higher number of new nodes compared to a node with no connection. The scale-free networks adhere to power-law distribution where the probability $P(i)$ of preferential attachment of a new gene to the existing gene i is given by

$$P(i) \sim d_i^\gamma + b \quad (2.50)$$

where d_i denotes the number of adjacent edges not initiated by gene i itself (which approximates to the in-degree of gene i). The parameters γ denotes the power of preferential attachment and b the attractiveness of the gene with no adjacent edges. New edges are added until the pre-specified number of edges is achieved. Since the GRN are causal, we randomly flipped the direction of edges to create feedback loops while generating the network [75].

Scale-free network topologies were generated with maximum in-degree $d_i = 5$ and $\gamma = 1$. GRNs defined by scale-free networks of known number of genes were simulated, and the parameters or the conditional probability distribution (CPD) tables $\{\theta_{ijk}\}_{i=1,j=1,k=1}^{I,J_i,K}$, were fixed for all bootstrapped samples. The genes were assumed to take three states $\{-1, 0, +1\}$ representing down-, no-, and up-regulation of genes. Genes with no parents were set to up-regulation with corresponding CPD values given by $= \{0.1, 0.1, 0.8\}$; For all remaining genes, the states was determined by the dominant state in parent set.

Table 2.1: Accuracy of building DBN-MCMC model with Dynamic Bayesian Networks in the presence of 10% Gaussian noise.

#Genes	#Time	Method	FP	FN	Precision	Recall	F-measure	Stability
10	10	MCMC	4	13	0.54	0.29	0.37	0.21
		GA	11	32	0.35	0.15	0.20	0.10
	20	MCMC	7	5	0.65	0.70	0.67	0.47
		GA	11	6	0.51	0.66	0.57	0.54
	50	MCMC	4	1	0.79	0.95	0.86	0.73
		GA	6	1	0.73	0.93	0.82	0.80
	100	MCMC	2	0	0.91	0.99	0.94	0.86
		GA	8	0	0.69	0.99	0.91	0.88
20	10	MCMC	17	29	0.36	0.24	0.29	0.12
		GA	11	32	0.35	0.15	0.20	0.10
	20	MCMC	25	17	0.46	0.55	0.49	0.29
		GA	23	20	0.43	0.45	0.44	0.30
	50	MCMC	15	4	0.69	0.89	0.77	0.62
		GA	6	1	0.73	0.93	0.82	0.80
	100	MCMC	8	0	0.83	0.99	0.90	0.80
		GA	13	0	0.74	0.99	0.85	0.86
50	20	MCMC	115	58	0.26	0.41	0.32	0.15
		GA	89	57	0.32	0.42	0.36	0.17
	50	MCMC	120	9	0.43	0.90	0.58	0.40
		GA	89	16	0.48	0.83	0.60	0.44
	100	MCMC	50	0	0.66	0.99	0.79	0.65
		GA	44	6	0.68	0.94	0.79	0.64
100	50	MCMC	369	51	0.28	0.74	0.40	0.24
		GA	231	38	0.40	0.81	0.54	0.32
	100	MCMC	205	0	0.48	0.97	0.65	0.47
		GA	128	8	0.59	0.95	0.73	0.56
	200	MCMC	43	0	0.81	0.99	0.89	0.79
		GA	38	0	0.84	0.99	0.91	0.81

2.9.1 Structure Learning

Simulations were performed to evaluate MCMC against GA algorithms in finding the optimal structure of DBN. Initialization of the structures was done using mutual information as in Section 2.5.1. The histograms of mutual information were plotted for each pair of genes. The peaks of the histogram were taken to determine the threshold for the presence of regulatory connections.

Larger networks and smaller time-samples converged slowly to the optimal structure, hence simulations were performed at different numbers of individuals (N) and generations (Q). For MCMC, the number of iterations for burn-in and

sampling were progressively increased from 1000 for 10 gene network and 100 time-samples up-to 10000 for the 100 gene network with 10 time-samples.

Figure 2.3: Illustration of convergence of the structure with (a) MCMC simulations and (b) GA for a network of 100 genes and 100 time samples

Similarly, for GA, different combinations of $N=500/1000/2000$ and $Q=500/2000/3000$ were used. The GA was assumed converged when the maximum number of generations is reached or if the score difference is below a certain value for 1000 consecutive generations [76]. Crossover probability was set to $p_c = 1$ and a small mutation probability of $p_m = 0.1$ was added. Local minima were avoided by maintaining a minimum similarity threshold of $th_s = 0.7$ in each generation. The geometric prior on structure was set with $q = 0.33$.

Fig. 2.3 shows convergence plots for 100 genes and 100 time-samples using MCMC and GA, lower value of likelihood in GA is due to normalization constant. While MCMC converges steeply in 2000 iterations the corresponding GA with population size 500 converges much faster in 1000 generations and has higher accuracy. The advantage of GA also is that it can parallel process the individuals in each generation.

Performance measures such as precision¹, recall², and F-measure³ were evaluated using true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), respectively.

2.9.2 Effects of Prior

We also studied the effects hyper parameters on the stability and performance of the network in figure 2.4 (a), increasing the pseudo count α to 50 caused a

¹Precision = $TP/(TP + FP)$

²Recall = $TP/(TP + FN)$

³F-measure = $2 \frac{Precision \times Recall}{Precision + Recall}$

profound increase in prediction suggesting that large networks have extreme probability distributions. In figure 2.4(b), the structural prior parameter q was gradually increased in GA. The optimal value seems to be 0.2 with a high F-measure (0.8). It showed a dip near 0.5 which is closer to a uniform prior. Increasing q further only reduced performance since a large number of parents can result in false positives in sparse graphs.

Figure 2.4: Effect of varying hyper-parameters for a network of 100 genes and 100 time samples on F-measure and Stability against (a) Dirichlet pseudo counts with $q = 0.2$ (b) Structural prior on average number of parents q with $\alpha = 1$

2.9.3 Effects of the Number of Time points

Given the size of the network being built, in order to study the effects of the number of time points, gene expression values were generated up to a time-span $T \in \{10, 20, 50, 100\}$ time points from GRN, assuming DBN as described in Algorithm 3. First, gene expression at the first time point were initialized randomly; second, gene expressions at other time points were sampled from DBN, using (2.46); and third, Gaussian noise with variance $\sigma^2 = 0.1$ were added. Structural learning was done using (i) MCMC simulations and (ii) GA. The convergence was determined when there was no change in fitness or generations reached 1000 iterations. For a given n number of genes and T number of time points, $B = 100$ bootstrap time-series datasets were generated.

Table 2.1 shows convergence and stability of building networks of different sizes with varying number of time points. As seen, to obtain decent accuracy and stability, the number of time points needed should approximately be equal to the number of genes in the network. Interestingly, our simulations indicate

that small networks required more number of time points, relative to the number of nodes in the networks, than the large networks. For shorter time-series, precision and recall measures were low, indicating that the performance was adversely affected by the presence of large number of false negatives (FN) and false positives (FP) (more by FP). Increase in the number of time points improved the overall performance; Note that the improvement was higher for larger networks. It is easier to correct false negatives by increasing the number of time points than the false positives.

In general, MCMC achieve higher precision at the cost of larger number of iterations. Performance of GA is better than MCMC when the number of time-samples is small and network size is large. For example for 100 genes and 50 time-samples a GA performed much better than MCMC.

Figure 2.5: Distribution of stability of edges of gene regulatory networks obtained using MCMC and GA (a) with 50 genes and 50 time samples and (b) with 100 genes and 100 time samples

2.9.4 Stability of the Networks

The last column of Table 2.1 shows that the stability of learning GRN structure by using DBN decreases with the number of time points relative to the network size. Fig. 2.5 compares the stability of individual edges of networks containing 50 and 100 genes by MCMC and GA. For 50 genes an MCMC has remarkably high stability while in a GA the edges have poor stability, however for 100 genes both methods had similar stability. The overall network stability of GA which is better than MCMC must be on account of the large number of lower false positives. Accuracy of networks with stability threshold set to 0.5 for

50 and 100 genes is given in Table 2.2. We infer that stable edges usually correspond to TP.

Table 2.2: Accuracy of derived networks derived with a stability threshold of 0.5 by using MCMC simulation and GA

#Genes,#Time	Method	FP	FN	Precision	Recall	F-measure
50,50	MCMC	0	0	1.0	1.0	1.0
	GA	69	9	0.56	0.91	0.69
100,100	MCMC	0	0	1.0	1.0	1.0
	GA	3	1	0.98	0.99	0.98

2.9.5 VAR and Gaussian Networks

To generate continuous synthetic time series we sampled random number from a normal distribution with zero mean and unit variance. Expression of a gene was weighted some of parent expressions in previous time points. The connectivity parameter was arbitrarily set to 0.5 and the sign was toggled between (+/-) among the parents. Table 2.3 compare the different models when data is generated from the model. It can be seen that MVAR models are very simple and hence easy to reconstruct from synthetic data. Gaussian models on the other hand are more complex than DBN and show several FP on synthetic data. The last column shows the estimated time in mins, DBN are extremely fast requiring less than a minute on networks with 10 nodes, MVAR takes about ten mins while GBN will only give good prediction after one hour. DBN can predict a network of ten nodes in just 30 seconds (0.5 min). While complexity increases linearly with network size, it was not significantly large on synthetic networks.

Table 2.4 compares performance of DBN to GBN when continuous synthetic data is used. It can be seen that DBN performs slightly better than GBN on the 10 gene network, and much better than the GBN on the 20 gene network. This, could be due to time-delay in the network or complexity of feedbacks and loops which are not well handled by a GBN. The DBN model also converged several times faster than GBN which is critical in large network prediction. Since, continuous data is closer to real time-series we can conclude that DBN is better suited to large networks.

2.9.6 Sieve and Residual Probabilistic Bootstrap

To bootstrap we used residual bootstrapping to generate more samples. A comparison was done of probabilistic residual bootstrapping and vector sieve

Table 2.3: Accuracy of networks derived using DBN, VAR and Gaussian BN for 10 genes when synthetic data is generated from the model. The running time in mins is also reported.

#Time	Method	FP	FN	Precision	Recall	F-measure	Speed (min)
50	DBN	4	1	0.73	0.93	0.86	0.25
	MVAR	1	2	0.94	0.88	0.9	5
	GBN	7	16	0.22	0.11	0.14	30
100	DBN	2	0	0.91	0.99	0.94	0.5
	MVAR	0	2	1	0.88	0.95	10
	GBN	5	13	0.5	0.27	0.35	60

Table 2.4: Accuracy of networks derived using DBN and Gaussian BN for 10 genes and 20 genes when continuous synthetic time series of length 200 was used.

#Genes	Method	FP	FN	Precision	Recall	F-measure
10	GBN	10	0	0.44	0.44	0.44
	DBN	7	3	0.53	0.44	0.48
20	GBN	70	31	0.1	0.18	0.13
	DBN	168	0	0.19	1	0.32

bootstrap where noise was added from error distribution. For vector sieve we used a 10 gene set and 30 time points 500 bootstraps were resampled. Table 2.5 shows that bootstrapping at a threshold of 0.4 can detect 4 more TP than the original time series and hence the recall is much higher. When residuals are sampled from original time-series of length 50, the bootstrap samples of length 30 can also recover the original network perfectly.

Next we used probabilistic bootstrap to resample residuals from 30 time points and 10 gene network. Bootstrap gave fewer false positives than DBN and Sieve bootstrap. We can hence conclude that probabilistic bootstrapping is optimal for GRN prediction. When CPT are specified using 50 time points. The bootstrap method can recover the network perfectly and has previously been shown in Table 2.5.

Table 2.5: Accuracy of vector sieve and residual bootstrap for 30 time points

#Model	Method	FP	FN	Precision	Recall	F-measure
VAR	Original	1	6	0.92	0.66	0.77
	Bootstrap	11	2	0.59	0.88	0.71
	Bootstrap(50)	0	0	1	1	1
DBN	Original	9	2	0.64	0.88	0.74
	Bootstrap	5	4	0.73	0.77	0.76
	Bootstrap(50)	0	0	1	1	1

2.9.7 Experiments on Real Datasets

Three real datasets were used to demonstrate the utility of our techniques : two datasets obtained in cell-cycle of yeast, and *M.tuberculosis* DNA repair pathway of *M. tuberculosis* and third one from human Hela cell cycle. All the data has been accumulated at equally spaced time points. The networks were obtained by using DBN and DBN with bootstrapping. Comparisons were performed against GRN and MVAR techniques.

2.9.7.1 Yeast Cell-Cycle Data

Numerous biological changes associated with cell-cycle make it an attractive model for the study of genome-wide regulation of gene activity. Moreover, abnormalities in cell-cycle regulation leads to many biological complications and diseases including cancer. Simultaneous identification of all the genes in a genome which coordinately regulate during the cell-cycle provides a consistent internal standard for comparison of gene activity over time. The cell cycle data has been collected on yeast cell-cycle on non-pathogenic cells. Because more than 20% of human proteins display significant homology to yeast proteins, our findings also link a range of human genes that are related to cell-cycle period-specific biological function [77]. Proliferation of all cells is mediated through cell cycle that consists of four main phases: genome duplication (S phase) and nuclear division (mitosis or M phase), separated by two gap phases (G1 and G2). The S-G1-M-G2-S forms a complete cycle of cell duplication.

Our methods were evaluated on time-series gene expressions of yeast cell-cycle data over 17 time points obtained from Chou et al. [78]. We refer to this dataset as Chou dataset. In this study, commercially available high-density oligonucleotide arrays were used to quantitate mRNA transcript levels in synchronized yeast cells at regular intervals during cell cycle. To obtain a synchronous yeast culture, *cdc28-13* cells were arrested in late G1 at start by raising the temperature to $37^{\circ}C$, and the cell cycle was reinitiated by shifting cells to $25^{\circ}C$. Cells were collected at 17 time points taken at 10 min intervals, covering nearly two full cycles of cell duplication. Cell cycle-dependent periodicity was found for 416 of the 6220 monitored transcripts. Cells exhibited 95% of synchrony throughout the time course, as determined by bud size and nuclear position.

To obtain cells synchronized in a different way, an isogenic strain bearing the *cdc15-2* allele, which enables arrest in the late G2, was used to generate a second time course. The 1384 genes whose normalized mRNA level changed

by more than 2-fold during this portion of the time course were visually examined for periodicity of expression in both *cdc28-13* and *cdc15-2* strains. 416 genes were thus identified, that demonstrated consistent periodic changes in transcript levels.

The time courses were divided into early G1, late G1, S, G2 and M phases based on the size of the buds, the cellular position of the nucleus, and standardization to more than 20 transcripts whose mRNA fluctuations have been previously reported. Bud size was determined as unbudded, small-budded or large-budded. Bud size as a percentage of mother cell size was calculated after measuring buds and mother cells from photographs by using a ruler. Because cells are analyzed over two cell cycles, postmitotic cells from the first division need to be reset at some point to a premitotic cell of the second division. The most appropriate time to make reset is at 110 min. The beginning of S phase was determined by the appearance of the buds and the end of S phase was determined by the attainment of the large budded state.

There were 134 of the 416 cell cycle-regulated transcripts peaked in late G1, while only 56 transcripts peaked during M phase. Transcripts that peaked in late G1 displayed particularly sharp rates of accumulation and decay, while transcripts that peaked in S generally displayed a less dramatic induction pattern.

We considered only nine genes (*Swi4*, *Swi6*, *Swi5*, *Fkh2*, *Mcm1*, *Ndd1*, *Cln3*, *Ace2*, *Mbp1*) from this data set to demonstrate the control of sequential activation of cyclins and other cell cycle regulators [10]. The group of genes *Swi6*, *Swi4* and *Mbp1* are active in phase late G1; genes *Mcm1*, *Fkh2*, *Ndd1* are active in phase G2/M; and genes *Swi5*, *Ace2*, *Mcm1* are active in phase M/G1; gene *Cln3* are active in phase late G1. A smaller subset of nine genes were used because validation with protein-protein interaction data is practically not possible for large datasets. Fig. 2.6 is used as the target network as reported in [10]. Arrows represent gene regulation relationships. Each group of genes regulate other genes acting in the next phase of the cell cycle. Preprocessing included noise removal, scaling and discretization to minimize mutual information.

2.9.7.2 *M.tuberculosis* DNA Repair Data

The methods were evaluated on *M.tuberculosis* DNA repair data obtained on tuberculosis that needs urgent attention and treatment because of the increasing number of infections and its drug-resistance. Target selection is critical to the development of new drugs but is hampered by the lack of understanding

Figure 2.6: Target network of 9 genes in the cell cycle dataset of Chou et. al

of the dynamics of metabolic responses of interruption of target function even by the current agents. Analysis of the transcriptional response by microarray can, in theory, provide clues to such adaptive responses, but thus far gene expression profiles have only been used to extensively study the mechanisms of action of a small number of related drugs.

Coordinately regulated sets of genes (regulons) are often controlled by a single transcriptional regulators that function as genetic master switches, committing the bacterium to a major alteration in metabolism. In this study, genome-wide expression profiling has been applied to diverse environmental changes, including many different drug types, to begin to map the adaptability of the bacilli for interruption of specific arms of metabolism.

The differential transcriptional response of *Mycobacterium tuberculosis* to drugs and growth-inhibitory conditions was monitored to generate a data set of 430 microarray profiles. Analysis of this data set further revealed 150 underlying clusters of coordinately regulated genes offering the first glimpse at the full metabolic potential of the organism. The underlying gene clusters further provide fundamental insights into the metabolic response of bacteria to drug-induced stress and provide a rational basis for the selection of critical metabolic targets for screening for new agents with improved activity against this important human pathogen [79] .

Microarray preparation is described under GEO accession number GSE1642. Expression ratios were calculated as the difference of feature pixel median and background pixel median for one color channel divided by the same for the

other channel. Expression ratios were transformed to the log base 2 for all further calculations. Standardized gene expression ratio patterns were calculated by subtracting the mean expression ratio and dividing by the standard deviation statistics calculated from all ratios (all microarrays) for that gene. Analysis of cellular response to translational inhibition revealed a general down-regulation of macromolecular synthesis, as expected, although there was an evident attempt to increase synthesis of the translational apparatus.

In our analysis, we focussed on the responses of bacteria to drug-induced stress. Treatment with mitocycin C caused DNA damage and hence led to upregulation of associated repair genes. Eight time points are available at NCBI Gene Expression Omnibus (GSE1642-GPL1396 series) database. The system consists of the proteins LexA and RecA as well as up to 40 genes that are regulated by these two proteins together. If a damage in the DNA that results in single-stranded DNA occurs, then the *M.tuberculosis* DNA repair system is activated. The protein RecA binds to this single-stranded DNA and produces a change of the structure of the protein LexA (Fig. 2.7). In this form, LexA cannot bind to the regulatory regions called SOS boxes on the DNA any longer. The associated SOS genes will be no longer repressed by LexA and transcription starts. Both genes *recA* and *lexA* have an SOS box as well. Additionally to the SOS regulation there exists another mechanism in *M. tuberculosis* to activate some of the SOS genes in the case of DNA damage [80]. In order to get insight into the unknown alternative mechanism we include one gene that is solely regulated by the *recAlexA*- mechanism, namely *linB*, and one gene, which is additionally regulated by an alternative mechanism, namely *ruvC* was checked in the analysis. In [81], the gene *Rv2719c* is predicted to play an important role in the *M.tuberculosis* DNA repair system.

Eight time points are available at NCBI Gene Expression Omnibus (GSE1642-GPL1396 series) 0.33hr, 0.75hr, 1.5hr, 2hr, 4hr, 6hr, 8hr and 12hr after DNA damage. We used 9 genes previously specified [23]. We combined replicates using component analysis to find the dominant component. The data was discretized using unsupervised division into 3 levels in Weka.

2.9.7.3 Human Hela Cell Cycle

The genome-wide profile of gene expression during the cell-division in a human cancer cell line (Hela) was characterized using DNA microarrays. Human cell cycle was studied by observing the genes whose expression changes when primary fibroblast cultures reenter the cell cycle from a serum-deprived resting

Figure 2.7: Target network of 9 genes in MTB dna repair pathway

state. Total of 700 genes were identified as cell cycle regulated in primary human fibroblasts.

Hela-53 cells were plated in tissue culture dishes. Cells were arrested in S phase. RNA was prepared from the cells collected typically at intervals from one to two hours. Mitotic cells were collected every 10 mins. Reference RNA was prepared asynchronously growing Hela cells. Spotted cDNA microarrays containing 16,300 human genes were manufactured in Stanford Inc. Spots of poor quality were removed by visual inspection. Only features with signal intensity atleast 20% above background and with good quality in 80% of the samples in a given time course were analysed further. Data points that did not meet these criteria were left blank in the primary table. Missing data was estimated using k-nearest neighbor with $k=12$.

Because of systematic differences between experiments (e.g, array batching, labelling methods, and synchronization methods), each time course was centered independently by filtering out the first most significant eigenvalue. The data has been previously normalized so that the average $\log_2$ ratio over the course of the experiment is zero. Fourier transform (FT) has been applied to the time-series of each gene. Next, expression profiles were correlated to five different known profiles corresponding to S, G2, M, G1 and M-G1 phases, by using a standard Pearson correlation. Corresponding profiles were identified by averaging log ratio of data. The peak correlation score was defined as the highest correlation value between time-series of each gene and each of the profiles. The FT was further scaled by the peak correlation value of the gene.

Here, we used a gene set of 51 genes earlier identified to be regulated by the cell cycle [75]. The data containing 48 time samples were discretized by using unsupervised division into 3 levels in Weka.

2.9.7.4 Bootstrap on Real data

We simulated the three dataset described in Chapter 1. Preprocessing and discretization were also performed. Here we use uninformative priors $\alpha_{ij} = q_i$ and $\alpha = 20$. We show validation of our results by employing BioGrid protein-protein interaction (PPI) data [34] marked as TP and edges with wrong directions marked as $\hat{TP}$. Table 2.6 shows the bootstrap confidence intervals for the three datasets as well as validation using protein interaction data. Some edges are identified with wrong directions shown as TP_D in the graphs. Direction of causality are known for benchmark networks from yeast cell-cycle and Tuberculosis DNA-repair pathways. For the larger network of human cancer pathway the directional information was not available hence bi-directional was assumed. It can be seen that the DBN has a high likelihood has higher true positives compared to the VAR method.

2.9.7.5 Yeast Cell-Cycle Data

For Chou pathway the DBN correctly detects self regulation of Swi4 which occurs with short-delay. It also predicts regulation of Swi5 in M phase by Ndd1 in G2 and regulation of Ndd1 by Swi4 in S phase. These interactions are observed in the S-G2-M phase spanning only 30 mins. Regulation of Mbp1 by Cln3 is also identified as both genes co-occur in G1 phase. The DBN also identifies other short-delay interactions such as between Ndd1 and Swi4, Swi5 and Fkh2, Cln3 and Ace2.

Overall the TP by DBN is 9 compared to only 7 given by VAR. The bootstrap likelihood interval is also higher for TP in DBN indicating a better fit to the data. Fig. 2.8 shows several interactions predicted using first-order Markov model. First order interactions seem to occur in the same phase in the target network. Bootstrapping has been done to assess confidence of edges and reduce the false positives. Dirichlet prior with pseudo count 20 was used for the yeast network.

2.9.7.6 *M.tuberculosis* DNA Repair Data

The first-order DBN could only recover 6 TP in the MTB pathway which is similar to the VAR model. True positives include activation of *lexA* by *linB*

Table 2.6: First- and higher-order regulations predicted by DBN(d) and VAR models built on datasets by Chou *et. al.* (9 genes), Tuberculosis (9 genes) and Hela cell line(51 genes). The number of genes in the dataset, (likelihood), total number of regulations, true positives with protein-protein interaction data. $\hat{TP}$ are edges predicted with incorrect direction and k -skip validation is using gene cascades of k genes.

#genes	model(o)	# edges at delays	Validation		likelihood
			TP	$\hat{TP}$	
Chou(9)		1			
	d(1)	17	4	5	[-139.39,-137.9]
	VAR	19	4	3	
Mtb(9)	d(1)	12	4	2	[-68.46,-66.42]
	VAR	10	4	2	
Hela(51)	d(1)	159	13	6	[-2085.78,-2079.16]
	VAR	111	2	6	

which is known to initiate the dna-repair pathway. Regulation of *ruvC* by *fadD23* which appears at the end of the repair also was correctly identified. However, DBN was unable to identify several time-delayed interactions. Confidence of dbn edges (Fig. 2.9) is much higher for true positives.

2.9.7.7 Human Hela Cell Cycle

The importance of time-delays becomes significant in regulatory networks, especially when a large number of genes are involved. In general, the first-order interactions are presumed to be instantaneous events such as initialization, silencing, etc.; the time-delayed interactions were mostly regulatory or feedback associated. The DBN could only predict 19 interactions correctly in the pathway compared to only 9 by VAR model suggesting that it is more suited to large networks. In Fig. 2.10 several new interactions are identified by the DBN for the Hela cell cycle. Since time-delayed information is not easily available for such a large network we looked at the top nine hubs conserved in the pathway.

The top hubs predicted by DBN and HDBN are compared with Biogrid in Table 2.7. The DBN identified hubs FGF1,FGF18 and C-Jun correctly. We looked at the function of these hubs and found that indeed they were instantaneous such as initiation of cell division by FGF or involved in apoptosis such as PUMA. C-Jun levels are constant in the cell-cycle indicating constant feedback and regulation.

Discussion and Conclusions

Biological networks are constantly subjected to random perturbations and noise, and efficient feedback and compensatory mechanism naturally provide

Figure 2.8: Time-delayed interactions in a predicted network using DBN bootstrap in yeast cell cycle.

stability. Gene expression data gathered from microarray are confounded by not only by biological noise and variations but also measurement errors and artifacts. Dynamic Bayesian networks (DBN) are widely used to build gene regulatory networks (GRN). We observe that large number of time samples can accurately predict GRN. However, due to noise and shortage of time samples make GRN built from time-series are often unstable or non-reproducible.

In this chapter, we investigated stability of building GRN in DBN framework by using synthetic data and consider two different heuristic based on MCMC and GA. By bootstrapping gene expression time-series datasets synthetically from scale-free networks (taken as topology closer to biological networks), stability of building GRN of different sizes, with varying number of

Figure 2.9: Time-delayed interactions in a predicted network using DBN bootstrap in *M.tuberculosis* DNA repair pathway.

Table 2.7: Top 9 hubs obtained for 51 genes in Hela cell line of human cell-cycle, ranked in the order of their connectivity. The hubs obtained from the BioGrid are also given for comparison.

model(<i>o</i>)	Rank of genes based on connectivity								
	1	2	3	4	5	6	7	8	9
BioGrid	FGF1	P53	FGF18	PUMA	BCL2	NOXA	BCL-XL	C-JUN	P21
d(1)	9	9	8	5	5	5	5	4	4
	PUMA	FGF1	FGF18	GADD45	CYCLINE1	C-JUN	RNAHP	A20	TRF4-2
	17	12	11	11	10	9	8	8	8

time points were studied. Although a number of time points equal to the number of genes in the network provide good accuracy, presence of false negatives and false positives were high. This could only be reduced by increasing the number of time points at which gene expressions are gathered. Stability of DBN to changes in parameters is high even in large networks.

While MCMC can achieve high accuracy with small number of time-samples it becomes computationally impossible as network size increases. A GA solution then can parallel process populations of structures with comparable performance. If the number of time-samples is much lower than network size a GA can give a reasonable structure where the MCMC fails to converge at all. Use of priors for parameters and structure can significantly improve prediction.

Figure 2.10: Time-delayed interactions in a predicted network using DBN bootstrap in Hela human cell cycle.

(a)

The work with synthetic data provides a foundation for future studies of stability of building GRN with real data. In practice, although gene expressions of a large number of genes are gathered simultaneously, repetition over many time points is limited due to high cost and time involved, and due to the nature of experiments. In such scenarios, either bootstrapping or regularization of time samples of gene expressions is needed to study the stability and reproducibility of computationally derived GRN. Investigation on real data reinforce our inference using synthetic data.

Chapter 3

Higher-order Dynamic Bayesian Networks (HDBN)

Biological networks are prone to delays in regulation. Genes in the pathways have variable synthesis as well as degradation rates, and feedback mechanisms maintain stability of complex molecular events [8]. In a first-order DBN the expression value of a gene is assumed to be dependant on parent gene expressions in the previous time-point. This assumption can be extended to longer delays in a higher-order (HDBN) where parent gene expressions in the previous two or three time points can also be considered. While a few authors have used HDBN to represent delays in regulation, the order or maximum delay of such networks has been fixed. To overcome this limitation, in this chapter, we introduce a variable-order DBN (VDBN), where the order of delay between genes and parents is automatically determined using a variable-order MCMC technique. We focus on the following aspects : (1) how VDBN can include higher-order delays by combining several layers of DBN and reducing the number of parameters by variable delay (2) how a reversible variable-order MCMC can determine the optimal order of an interaction efficiently (3) we look at the stability of the networks and the delays (4) how probabilistic bootstrapping can be extended to higher-order DBN so as to assess the confidence of the connections and their corresponding delays (5) lastly, we validate the model using synthetic as well as real data. To validate delays we use cascades of first-order protein-protein interactions (PPI) [36].

Similar to chapter 2 we tested our model on synthetically generated networks with delays as well as real time series data obtained in the yeast cell cycle, M. tuberculosis and human Hela cell-cycle. The new acceptance mechanism allows reaching a structure that renders the optimal order or the maximum delay of regulation of genes. We also compared the networks predicted with those given by DBN in chapter 2. Several new interactions and corresponding

delays were correctly predicted by VDBN. In order to validate higher delay interactions we look at cascades of genes in the GRN, which together correspond to an interaction in a PPI network. A comparison of key genes and hubs found in the network and relevance of delays to nature of activity in biological processes was done.

The chapter first describes the architecture for higher-order delays as an extension of DBN by introduction of two or more hidden layers of nodes [82]. We explain the parameter estimation where each parent can have only a single time delay while regulating a gene hence remarkably reducing the number of parameters. We extend the topology prior on number of parents to a connectivity prior on delays [83], [84]. Next, we present a reversible jump MCMC to learn the structure of the model. A detailed balanced condition allows the MCMC to jump effectively between parameter sub-spaces of different delays [85]. The residual probabilistic bootstrap is implemented for the case of delays to generate additional samples and determine the confidence of estimated edges and delays. Lastly, indices for stability of the model and stability between delays are derived.

3.1 Motivation and Contributions

The first-order assumption that the expression value of a gene depends only on the gene expressions of previous time points in DBN makes, regulatory interactions synchronous or are considered to take place after a delay of time between two time points. Biological networks are inherently synchronous and therefore DBN are insufficient to model more complex regulatory mechanisms of molecular networks [86]. For example, p53 gene expression is inhibited by over expression of gene mdm2 and forms a feedback loop in the system. A model with delays was studied in [49] and found to be more accurate than one without delays.

Previous attempts in building gene regulatory networks including delays were based on mutual information (MI) between the expression values [10], [87]. Such networks find only functionally correlated genes but fail to provide causal interactions of regulatory networks. Granger causality can detect functional connectivity from time series data where optimal order is determined using regression [88], [89]. The method works well on only certain small benchmark networks with low order delays. Context trees have been also used to model time-series data as it takes the form of a variable-order Bayesian network and has many fewer parameters but is unable to model feedbacks which are realistic

in biological networks [90]. The recent attempts to build GRN including delays with higher-order DBN (HDBN) need the order to be fixed before determining the structure of the network [10], [91].

Delays in pathways may be due to environmental disturbances or during assembly of intermediate products. A structure with more memory states can represent the information in these delays. However, this comes at the expense of number of parameters. The number of parameters in a higher-order model increases exponentially with delays. By restricting interactions to have a single optimal delay we can greatly reduce the complexity of the model. The traditional MCMC is extremely slow in reaching global optima as it samples from a joint posterior distribution of all the parameters in a model with fixed dimensionality. A reversible jump MCMC (RJMCMC) is more flexible as it is able to move between models of different dimensionality based on context [12]. The detailed balanced condition between an increase in delay and decrease in delay ensures a stationary distribution [92]. RJMCMC has been previously used to determine topology of GRN, here we apply it to find the optimal delay as well as topology [84].

Analogous to the topology prior for number of parents we have a connectivity prior for increasing or decreasing the delay based on mutual information between genes at different delays. Probabilistic residual bootstrap allows us to determine the confidence and stability of delays. An index for stability of delay over the bootstrap samples is developed. Validation of our methods is done in two ways (1) by using synthetic data with delays and (2) comparison with known biological knowledge in the corresponding protein interaction data. Since, information on delays is not available for many interactions we resort to cascades of known interactions to validate delays. Key genes and hubs are discussed and advantages of VDBN over DBN are highlighted.

3.2 Parameter estimation in HDBN

A Bayesian network can model the joint likelihood of a set of random variables by taking into account the conditional independencies of the variables. Dynamic Bayesian networks (DBN) extends the Bayesian network framework to capture the temporal characteristics of gene expressions by assuming a first-order stationary temporal process. Suppose that time courses of expressions of n genes are given by $x = (x_i(t))_{i=1, t=1}^{n, T}$ where $x_i(t)$ denotes the value of the expression of gene i at time t and T denotes the total number of time points which are assumed to be equally spaced.

Figure 3.1: Representation of second-order DBN and its connectivity matrix.

Given the structure s of the transition network, a Bayesian network decomposes the likelihood of gene expressions in an array into a product of conditional probabilities by assuming independence of non-descendant genes, (3.1)

$$p(x|s, \theta) = \prod_{i=1}^n p(x_i|a_i, \theta_i) \quad (3.1)$$

where $p(x_i|a_i, \theta_i)$ is the conditional probability of gene expression $x_i = (x_i(t))_{t=1}^T$ given the set of parents' gene expressions, a_i , where θ_i denotes the parameters of the conditional probability. The structure s is defined by the order r and the parameter set $\theta = (\theta_i)_{i=1}^n$.

Higher-order DBN (HDBN) is a r -order DBN ($r > 1$) that assumes r -order Markov dependence of gene expressions over time. An HDBN can model regulations with different time delays in the network. The structure of an r -order HDBN is represented by a connectivity structure comprising of $(r + 1)$ consecutive time points and n genes, or a graph of $(r + 1) \times n$ nodes. The structure of the network is represented in a matrix $c = \{c_{i,j}\}_{n \times n}$ where $1 \leq c_{ij} \leq r$ denotes the delay of regulation of gene i by its parent gene j . One unit of the delay is equal to the time interval between two consecutive measurements of gene expressions. We presume that one gene regulates another gene with at most with one single time delay. The regulations are assumed to be stationary. The full temporal characteristics of genes are represented by unrolling the transition network over time.

The equation for higher-order DBN is a simple extension of that for DBN in (2.3) where probability of genes is dependent on parents in time $t, t-1, \dots, t-r$

where r is maximum delay or order of HDBN. The r -order Markov model can then be described by (3.2)

$$p(x|s, \theta) = \prod_{i=1}^n p(x_i^t | a_i^t, \theta_i^t) \prod_{r'=1}^r \prod_{i=1}^n p(x_i^t | a_i^{t-r'}, \theta_i^{t-r', t}) \quad (3.2)$$

where $x = (x_1, x_2, \dots, x_n)$ denotes the vector of gene expressions of n genes, $p(x_i^t | a_i^{t-r'}, \theta_i^{t-r', t})$ denotes additional parameters for the conditional probability of gene expression x_i given its parent gene expressions a_i from previous r time points, and $\theta_i^{t-r, t}$ denotes the parameters of the conditional probabilities of the transition networks.

It can be seen that there is a different parameter set for each delay up-to order r of the HDBN. Similar to a DBN this is achieved by adding additional nodes for each time delay resulting in $(r + 1) \times n$ nodes, and input data is shifted by r' time points while modeling nodes of delay r' . This works well for lower order delays and small networks. However, as the model order r increases or the number of genes n becomes large and it is not possible to reliably estimate the parameters especially from short time-series.

Suppose that gene expressions are discretized into d levels such that $x_i(t) \in \{1, \dots, d\}$. Let the number of states that the set of parent gene expressions a_i can take be q_i ; and the set of different time delays, which the parent state j of q_i can take be l_j ($1 \leq l_j \leq r$). Thus, $q_i = d^{|a_i|}$ where $|a_i|$ is the number of parents of gene i , and the maximum number of delay combinations, which the parent set $|a_i|$ of gene i can take $w_j = r^{|a_i|}$. Let $\theta_{ijzk} = p(x_{i,t} = k | a_i = j, l_j = z)$ and by using the property of decomposability, the likelihood can be written as

$$p(x|s, \theta) = \prod_{i=1}^n \prod_{j=1}^{q_i} \prod_{z=1}^{w_j} \prod_{k=1}^d \theta_{ijzk}^{N_{ijzk}}. \quad (3.3)$$

Marginalizing (3.3) over the parameters:

$$p(x|s) \propto \int p(x|s, \theta) p(\theta|s) d\theta \quad (3.4)$$

where $p(\theta|s)$ denotes the prior densities of the parameters. Assuming that the densities of conditional probabilities are independent:

$$p(\theta_k|s) = \prod_{i=1}^n \prod_{j=1}^{q_i} \prod_{z=1}^{w_j} p(\theta_{ijzk}) \quad (3.5)$$

Substituting (3.3) and (3.5) in (3.4), we get

$$p(x|s) = \prod_{i=1}^n \prod_{j=1}^{q_i} \prod_{z=1}^{w_j} \int \prod_{k=1}^d \theta_{ijzk}^{N_{ijzk}} p(\theta_{ijzk}) d\theta_{ijzk} \quad (3.6)$$

We assume Dirichlet priors for the conditional densities:

$$p(\theta_{ijzk}) = \frac{\Gamma(\sum_k \alpha_{ijzk})}{\prod_{k=1}^d \Gamma(\alpha_{ijzk})} \prod_{k=1}^d \theta_{ijzk}^{(\alpha_{ijzk}-1)} \quad (3.7)$$

where hyper-parameters α_{ijzk} correspond to the parameters of the prior density θ_{ijzk} . $\theta = \{\theta_{ijzk}\}_{n \times q_i \times w_j \times d}$. The Dirichlet priors enable us to model complex interactions among the genes in regulatory networks. Substituting (3.7) in (3.6)

$$p(x|s) = \prod_{i=1}^n \prod_{j=1}^{q_i} \prod_{z=1}^{w_j} \frac{\Gamma(\alpha_{ijz})}{\Gamma(\alpha_{ijz} + N_{ijz})} \prod_{k=1}^d \frac{\Gamma(\alpha_{ijzk} + N_{ijzk})}{\Gamma(\alpha_{ijzk})} \quad (3.8)$$

where $N_{ijzk} = \sum_{t=1}^{T-r} \delta(x_{i,t} = k, a_i = j, w_j = z)$, $N_{ijz} = \sum_{k=1}^d N_{ijzk}$, and Γ denotes the gamma function. Here we take $\alpha_{ijz} = 1.0/(q_i w_j)$.

Using Lagrange theory, the maximum likelihood (ML) estimates of the parameters can be derived from (3.8):

$$\hat{\theta}_{ijzk} = \frac{\alpha_{ijzk} + N_{ijzk} - 1}{\alpha_{ijz} + N_{ijz} - d} \quad (3.9)$$

3.3 Variable-Order MCMC (VMCMC)

The order r of DBN is learned from data in a variable-order DBN (VDBN). Let the structure $s = (r, c)$ where c explicitly indicates the connectivity matrix indicating the delays of connections; and r denotes the maximum delays. In this section, we propose a variable-order Markov chain Monte Carlo (VMCMC) technique to find the maximum a posteriori (MAP) estimation of the structure of the network. The MAP estimation of the order (r^*, c^*, θ^*) is given by

$$(r^*, c^*, \theta^*) = \arg \max_{r, c, \theta} p(r, c, \theta | x) = \arg \max_{r, c, \theta} p(r, c, \theta, x) \quad (3.10)$$

where $p(r, c, \theta, x) = p(r)p(c|r)p(\theta|c)p(x|r, c, \theta)$.

The optimal structure is achieved by VMCMC simulation that form a Markov chain of structures and converge to an equilibrium structure. The

change in the order results in a change of the dimensionality of space of structure. Therefore, we propose a novel method that is capable of jumping between these subspaces and the structures within the subspace. The Metropolis-Hastings method of VMCMC is adopted, which associates an acceptance mechanism when the new sample structures are drawn. The acceptance of the new structure (r', c') is given by

$$\min \left\{ 1, \frac{p(r', c', \theta', x)}{p(r, c, \theta, x)} \right\} \quad (3.11)$$

where the acceptance ratio:

$$\frac{p(r', c', \theta', x)}{p(r, c, \theta, x)} = \frac{p(r')}{p(r)} \cdot \frac{p(c'|r')}{p(c|r)} \cdot \frac{p(\theta'|c', r')}{p(\theta|c, r)} \cdot \frac{p(x|\theta'|c', r')}{p(x|\theta|c, r)}. \quad (3.12)$$

The variable-order MCMC procedure allows the order of the GRN structure to change stochastically. In each iteration, a proposal mechanism probabilistically accepts the proposed order and the connectivity of the structure. Three types of changes to the order are proposed: (a) *increase* of the order; (b) *decrease* of the order; and (c) *no-change* of the order. Any change of the order is associated with a probability: let γ_+ , γ_- , and γ_0 be the probabilities associated with increasing, decreasing, and no-change of order, respectively, and then $\gamma_+ + \gamma_- + \gamma_0 = 1.0$. That is, if the current order is r :

$$r' = \begin{cases} r + 1, & p(r'|r) = \gamma_+ \\ r - 1, & p(r'|r) = \gamma_- \\ r, & p(r'|r) = \gamma_0 \end{cases} \quad (3.13)$$

The first term on the right hand side in (3.12) is the prior ratio of the orders. The prior of the network topology of order r is the product of prior probability of the network topology and the priors of the location of r nodes. The probability of selecting r -ordered subset from T -ordered points is $T^{-r}r!$. Therefore, the prior ratio for the increasing step is given by:

$$\frac{p(r')}{p(r)} = \frac{p(r+1)}{p(r)} \cdot \frac{r+1}{T} \quad (3.14)$$

3.3.1 Detailed Balanced Condition

The MCMC techniques needs to satisfy the detailed balance condition to achieve a convergence probability. To achieve the detailed balance with respect to $p(r'|r)$, if for all r, r' :

$$p(r)p(r'|r) = p(r')p(r|r') \quad (3.15)$$

This ensures a balance in dimensionality as we jump between subspaces of different orders.

In order to achieve the detailed balanced condition, in the increasing step, let $\gamma_+p(r) = \gamma_-p(r+1)$. This ensures that the Markov chain reaches an equilibrium state. From (3.11)-(3.15), the acceptance mechanism is given by

$$\min \left\{ 1, \text{prior ratio} \times \text{likelihood ratio} \times \frac{T - z(r)}{T} \right\} \quad (3.16)$$

the last term is the prior for changing order and $z(r)$ is the number of candidate delays that are not plausible [93]. Since increase in order is done by only adding new nodes to the network the Jacobian remains 1. Similarly, the acceptance mechanisms for movement of decreasing and no change of the order can be derived. For the decreasing step, the fraction of the above expression inverts. And for no change of order, only the likelihood ratio appears.

The MCMC algorithm to derive variable-order DBN is described in Algorithm 1. The structure is initialized using the mutual information between the gene expressions measured at different time points as described in [10]. In each iteration, the algorithm randomly selects and accepts a change of the order and the new structure, according to the acceptance mechanism in (3.11).

Algorithm 6 Variable-order DBN

```

 $r = 1$ 
Initialize topology of  $s$  using MI
Fix  $U$  as the number of iterations
 $u = 1$ 
repeat
  Choose  $\alpha = \text{rand}(0, 1)$ 
  Randomly select a gene with parent order  $r$ 
  Decide randomly on the new order  $r'$ 
  if  $\alpha < \frac{p(r', \theta', x)}{p(r, \theta, x)}$  then
     $(r, \theta) = (r', \theta')$ 
  end if
   $u \leftarrow u + 1$ 
until  $u \leq U$  or convergence

```

3.3.2 Stability of Delays

We propose a similarity based stability criterion for evaluating the predicted delays in GRN building algorithms, which is measured by the mean over all pairwise Hamming distances among all the detected connections and their delays over different datasets.

Let $\{x^b\}_{b=1}^B$ be a set of B samples of gene expression data and s^b be the GRN derived from b -th dataset x^b . Consider two GRNs with connectivity matrices $c^b = \{c_{i,i'}^b\}_{n \times n}$ and $c^{b'}$ derived from datasets x^b and $x^{b'}$, respectively. The similarity of two networks is obtained by the average Hamming distances over all regulatory connections:

$$\rho(c^b, c^{b'})_{d_1, d_2} = 1 - \frac{1}{N^b + N^{b'}} \sum_{i=1}^I \sum_{i'=1}^I d(c_{i,i'}^b, c_{i,i'}^{b'})_{d_1, d_2}. \quad (3.17)$$

where d_1, d_2 denotes the Hamming distance between networks with either delay d_1 or delay d_2 and N^b denotes the number of connections in network s^b have connectivity matrix c^b . The Hamming distance takes consider both the presence (1) and the absence (0) of regulatory connections between two networks. The stability of the network is in the range of $[0, 1]$ where a higher value denotes a higher stability of GRN inference algorithm.

Using the stability of two networks in (3.17), the stability of the GRN structure building algorithm is obtained by averaging over B number of structures. By averaging pairwise stability given in (3.17), the stability of the network structure is obtained as:

$$\rho_{\text{structure}} = \frac{2}{B(B-1)} \sum_{b=1}^B \sum_{b'=b+1}^B \rho(c^b, c^{b'})_{d_1, d_2}. \quad (3.18)$$

The delay distance of an independent connection for two delays d_1 and d_2 would be 1 if both edges had one of the two delays and 0 otherwise. It can be defined as

$$d(c_{i,i'}^b, c_{i,i'}^{b'})_{d_1, d_2} = \begin{cases} 1 & \text{if } c_{i,i'}^b, c_{i,i'}^{b'} \in \{d_1, d_2\}; \\ 0, & \text{otherwise.} \end{cases} \quad (3.19)$$

3.3.3 Probabilistic Residual Bootstrap

Similar to chapter 2 we extend the probabilistic residual bootstrap technique to higher order DBN. Here, also we resample residual from the error distributions and generate additional time series from the original. The likelihood of gene expressions $x(t)$ of genes at t is given by

$$p(x(t)|s, \theta) = \prod_{i=1}^I p(x_i(t)|x(t-r), \theta_i) \quad (3.20)$$

and the gene expression value of gene i is estimated by

$$\hat{x}_i(t) = \arg \max_k p(x_i(t) = k | \theta_{ijk}, x(t-r) = j) \quad (3.21)$$

where r is the order of the HDBN.

If the gene expressions at $t = 1$ is initialized the gene expressions, $\{x_i(1)\}_{i=1}^I$ at subsequent time points are generated by sampling from GRN by using (3.21).

If ML parameters $\{\theta_{ijk}\}_{i=1, j=1, k=1}^{n, q_i, d}$ are first specified, it is possible to determine the corresponding error probabilities $\{\epsilon_{ij}\}_{i=1, j=1}^{n, q_i}$ from the data [71]. By assuming that expressions of all genes are quantized into an equal number of levels the counts of residual gene expression are obtained as

$$\epsilon_{i, \hat{x}_i(t)} = \delta(\hat{x}_i(t) \neq \arg \max_k \theta_{ijk} | a_i(t-r) = j) \quad (3.22)$$

Assuming that the errors are time-independent, the error counts M_{ijk} and corresponding error probability can be computed as

$$\epsilon_{ij} = \frac{M_{ijk}}{T} \text{ where } M_{ijk} = \sum_{t=1}^T \epsilon_{ijk} \forall k \quad (3.23)$$

For a given short time-series we predict s and θ using BDe metric of DBN. The structure s is defined by a connectivity matrix $c = \{c_{ii'}\}_{n \times n}$ where $c_{ii'} = 0$ or $\neq 0$ depending on the presence or absence of a regulatory connection between two genes, i and i' . Next, B bootstrap samples are generated using θ and random errors are introduced from distributions ϵ_{ij} according to Algorithm (7).

Algorithm 7 Bootstrapping time series

Given: data $x(t)$, structure s , CPD θ and error distributions ϵ :

for $t = 1$ to T **do**

for $i = 1$ to n **do**

$a_i(t-1) = (x_{i'}(t-1)c_{ii'})_{i'=1}^n$

$\epsilon_i(t) = \text{sample}\{\epsilon_{ij} | j = a_i(t-r)\}$

$x_i(t) = \arg \max_k \theta_{ijk}$

if $\text{rand}[0, 1] \leq \epsilon_i(t)$ **then**

$x_i(t) = \text{rand}\{k\}_{k=1}^d$

end if

end for

end for

Confidence interval similar to chapter 2 can then be estimated from the bootstrap.

3.4 Experiments with Synthetic Data

Scale-free network topologies were generated with in-degree d_i set to a maximum of 5 genes and $\gamma = 1$. Since GRN are causal we randomly flipped the direction of edges to create feedback loops while generating the network [75]. Delays of order 2 were randomly added to the edges. GRNs defined by scale-free networks of 10 genes were simulated and the parameters, or the conditional probability distribution (CPD) tables $\{\theta_{ijk}\}_{i=1, j=1, k=1}^{I, J_i, K}$, were fixed for all bootstrapped samples. The genes were assumed to take three states $\{-1, 0, +1\}$ representing down-, no-, and up-regulation of genes. Genes with no parents were set to up-regulation with corresponding CPD = $\{0.1, 0.1, 0.8\}$; for all remaining genes the state was determined by the dominant state in parent set.

3.4.1 Structure Learning

Simulations were performed to evaluate VMCMC in finding the optimal structure of HDBN. Initialization of the structures were done using mutual information. The histograms of mutual information were plotted for each pair of genes. The peaks of the histogram were taken to determine the threshold for the presence of regulatory connections.

MCMC was used for structural learning with 20000 iterations for convergence for networks of size 10, 20 and 100 and for up-to four time delays. A Dirichlet prior was used for smaller networks however prediction was better without a prior for the 100 gene HDBN.

Performance measures such as precision¹, recall², and F-measure³ were evaluated using true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), respectively.

3.4.2 Effects of the Number of Time points

Given the size of the network being built, in order to study the effects of the number of time points, gene expression values were generated up to a time-span $T \in \{50, 100, 200\}$ time points from GRN, assuming HDBN as described in Algorithm 7. First, gene expression at the first time point were initialized randomly; second, gene expressions at other time points were sampled from HDBN, using (3.20); and third, Gaussian noise with variance $\sigma^2 = 0.1$ were

¹Precision = $TP/(TP + FP)$

²Recall = $TP/(TP + FN)$

³F-measure = $2 \frac{Precision \times Recall}{Precision + Recall}$

added. Structural learning was done using VMCMC simulations. The convergence was determined when there was no change in fitness or generations reached 20000 iterations. For a given I number of genes and T number of time points, $B = 100$ bootstrap time-series datasets were generated. We compared the use of VMCMC for variable-order DBN on a network of size 100 and up-to five time delays when 100 time samples are used. It can be seen that VMCMC gave fewer FP. The precision of the network was much higher at 0.81 when VMCMC is used compared to only 0.74 when MCMC is used. The network also converges much faster with a prior based on mutual information.

Table 3.1: Accuracy of networks derived using VMCMC and MCMC when five time delays are considered on 100 genes and 100 time points

Method	FP	FN	Precision	Recall	F-measure
VMCMC	23	88	0.81	0.55	0.67
MCMC	38	86	0.74	0.55	0.64

Table 3.2 shows convergence and stability of building networks of different sizes with varying number of time points and for delays two to four. As seen, to obtain decent accuracy and stability, the number of time points needed should approximately be equal to the number of genes in the network. Interestingly, our simulations indicate that small networks required more number of time points, relative to the number of nodes in the networks, than the large networks. For shorter time-series, precision and recall measures were low, indicating that the performance was adversely affected by the presence of large number of false negatives (FN) and false positives (FP) (more by FP). Increase in the number of time points improved the overall performance; Note that the improvement was higher for 20 gene network compared to 10 genes however a decrease was seen with 100 genes. It is easier to correct false negatives by increasing the number of time points than the false positives.

It can be seen that FN are higher for longer delays for example 20 genes network has 10 order 4 FN when simulated with 50 time points. This suggests that longer delays are more difficult to predict in the HDBN compared to smaller delays. Similar to chapter 1 the accuracy of the HDBN decreases with increasing order and lower number of time points. The stability of the network is almost half when using 50 time points compared to 100 time points. Accuracy of 100 gene networks is much lower in higher orders though networks are stable.

Table 3.2: Accuracy of building HDBN-MCMC model with Dynamic Bayesian Networks in the presence of 10% Gaussian noise.

#Genes	#Time	Method	FP(1,2,3,4)	FN(1,2,3,4)	Precision	Recall	F-measure	Stability
10	2	50	5(2, 3)	3(2, 1)	0.78	0.87	0.82	0.68
		100	2(1, 1)	3(2, 1)	0.94	0.89	0.90	0.83
	3	50	8(2, 3, 3)	4(1, 1, 2)	0.65	0.81	0.72	0.57
		100	3(1, 1, 1)	2(1, 0, 1)	0.92	0.92	0.92	0.83
	4	50	8(2, 2, 2, 2)	5(2, 1, 1, 1)	0.66	0.84	0.74	0.53
		100	4(1, 1, 1, 1)	4(1, 1, 1, 1)	0.88	0.88	0.88	0.78
20	2	50	12(6, 6)	20(9, 11)	0.6	0.5	0.5	0.47
		100	2(1, 1)	17(7, 10)	0.92	0.55	0.67	0.81
	3	50	21(7, 8, 6)	21(8, 5, 8)	0.5	0.5	0.5	0.41
		100	3(1, 1, 1)	17(6, 4, 7)	0.88	0.53	0.66	0.77
	4	50	25(5, 6, 9, 5)	20(2, 6, 3, 9)	0.4	0.44	0.44	0.34
		100	4(1, 1, 1, 1)	17(1, 5, 3, 8)	0.84	0.54	0.65	0.75
100	2	100	89(42, 47)	148(83, 65)	0.34	0.24	0.28	0.19
		200	132(67, 65)	139(80, 59)	0.3	0.29	0.28	0.20
	3	100	104(25, 49, 30)	172(63, 53, 56)	0.18	0.12	0.14	0.24
		200	134(30, 68, 36)	147(57, 47, 43)	0.22	0.19	0.21	0.5
	4	100	367(30, 22, 79, 236)	180(43, 42, 53, 42)	0.04	0.07	0.05	0.18
		200	306(31, 41, 65, 169)	172(39, 41, 50, 42)	0.07	0.11	0.08	0.34

3.4.3 Stability of the Networks

The last column of Table 3.2 shows that the stability of learning GRN structure by using VDBN decreases with the number of time points relative to the network size. For 20 genes an MCMC has remarkably high stability while however for 100 genes both methods had poor stability. The stability between delays of length one, two or three is shown in Table 3.3. The pattern is similar for smaller network of 10 genes and bigger network of 100 genes where longer delays have higher stability. However, the network with 20 genes shows a decline in stability for order four delays. This can be related to the results that bigger networks converge faster than smaller ones. Zero delay stability implies the edge has the same delay in both networks being compared and is usually low.

3.4.4 VAR Networks

Table 4.4 shows a comparison of VAR with HDBN model for network of size 10 and 100 time points. The VAR with delays on the 10 gene network performs very poorly confirming the non-linear nature of delayed pathways.

3.4.5 Vector Sieve Bootstrap

To bootstrap we used residual bootstrapping to generate more samples. A comparison was done of probabilistic residual bootstrapping and vector sieve bootstrap where noise was added from error distribution. For vector sieve we used a 10 gene set with delays up to order three and 50 time points 500

Table 3.3: Stability between delays of length zero, one, two and three when building HDBN-MCMC model with Dynamic Bayesian Networks in the presence of 10% Gaussian noise.

#Genes	#Time	Method	0	1	2	3
10	2	50	0.46	0.94		
		100	0.47	0.95		
	3	50	0.29	0.58	0.63	
		100	0.31	0.60	0.69	
	4	50	0.2	0.38	0.42	0.55
		100	0.23	0.42	0.47	0.61
20	2	50	0.45	0.93		
		100	0.47	0.95		
	3	50	0.29	0.55	0.69	
		100	0.31	0.58	0.74	
	4	50	0.22	0.46	0.43	0.32
		100	0.23	0.50	0.47	0.39
100	2	100	0.45	0.93		
		200	0.44	0.92		
	3	100	0.29	0.62	0.58	
		200	0.30	0.63	0.56	
	4	100	0.19	0.37	0.40	0.54
		200	0.21	0.38	0.42	0.57

Table 3.4: Accuracy of networks derived using VAR and HDBN when three time delays are considered on 10 genes and 100 time points

Method	FP	FN	Precision	Recall	F-measure
VAR	5	2	0.66	0.83	0.74
GBN	5	2	0.66	0.83	0.74
HDBN	0	5(2, 0, 3)	1	0.7	0.83

bootstraps were resampled. Table 3.5 shows that bootstrapping at a threshold of 0.5 predicts 18 less FP than the original time series and hence can correctly assess confidence of edges. When residual are sampled from original time-series of length 500, the bootstrap samples of length 50 can also recover the original network perfectly.

Next we used probabilistic bootstrap model to resample from 50 time points and 10 gene network with delays up to order three. As explained in motivation, it is difficult to perform bootstrap on short time-series data. When time-delays are included the sieve bootstrap fails to recover edges from time-series in VAR model. The probabilistic bootstrap can be used as a convenient method of sampling to check the confidence of edges and is scalable to large networks and delays. Bootstrap gave fewer false positives than HDBN and Sieve bootstrap. We can hence conclude that probabilistic bootstrapping is optimal for GRN prediction. When CPT are specified using 100 time points. The bootstrap method can recover the network perfectly and has previously been shown in Table 3.5.

Table 3.5: Accuracy of vector sieve and residual bootstrap for 50 time points and upto three time point delays

#Model	Method	FP	FN	Precision	Recall	F-measure
VAR	Original	28	6	0.17	0.5	0.26
	Bootstrap	10	9	0.22	0.25	0.23
	Bootstrap(500)	0	0	1	1	1
HDBN	Original	7(1, 2, 4)	7(2, 1, 4)	0.61	0.61	0.61
	Bootstrap	4(1, 0, 3)	7(2, 1, 4)	0.73	0.61	0.66
	Bootstrap(200)	0	0	1	1	1

3.4.5.1 Bootstrap on Real data

We simulated the three datasets described in Chapter 1. Preprocessing and discretization were also performed. Here we use uninformative priors $\alpha_{ij} = q_i$ and $\alpha = 20$. We show validation of our results by employing BioGrid protein-protein interaction (PPI) data [34] marked as TP and edges with wrong directions marked as $\hat{TP}$. Table 3.6 shows the bootstrap confidence intervals for the three datasets as well as validation using protein interaction data. Some edges are identified with wrong directions shown as TP_D in the graphs. A k-skip validation for time-delayed interactions is also performed. It can be seen that the HDBN has a higher likelihood than corresponding DBN and also much higher true positives.

3.4.5.2 Yeast Cell-Cycle Data

For Chou pathway the DBN correctly detects self regulation of Swi4 which occurs with short-delay. It also predicts regulation of Swi5 in M phase by Ndd1 in G2 and regulation of Ndd1 by Swi4 in S phase. These interactions are shown to occur in the S-G2-M phase spanning only 30 mins. Regulation of Mbp1 by Cln3 is also identified as both genes co-occur in G1 phase. The DBN also identifies other short-delay interactions such as between Ndd1 and Swi4, Swi5 and Fkh2, Cln3 and Ace2. However, we can see that it is unable to identify the longer-delay regulations such as those between Mcm1 and Swi6 from M to G1 phase which should occur over 60 mins time interval.

In order to detect the missing interactions we used a second-order VDBN on the yeast cell cycle data. Several new TP were found in addition to those predicted by DBN. In particular second-order TP include regulation of Cln3 by Ace2 and Swi5 during the long G1 phase. Several other interactions such as Ace2 with Fkh2, Mcm1 with Cln3 etc are also predicted though the direction of causality is incorrect. Overall the TP by second order VDBN is 14 compared to only 9 given by DBN. Using k -skip validation for delayed interactions could recover all 20 interactions in the benchmark network. 16 edges

predicted were from second-order emphasizing the differential gene expression in different stages of the cell cycle. The bootstrap likelihood interval is also higher in VDBN compared to DBN indicating a better fit to the data. Fig. 3.2 shows several time-delayed interactions predicted as second order marked as red. First order interactions seem to occur in the same phase in the target network. Bootstrapping has been done to assess confidence of edges and reduce the false positives. Dirichlet prior with pseudo count 20 was used for the yeast network.

Table 3.6: First- and higher-order regulations predicted by DBN(d), HDBN(h), and skip-chain(d:h) models built on datasets by Chou *et. al.* (9 genes), Tuberculosis (9 genes) and Hela cell line(51 genes). The number of genes in the dataset, (likelihood), total number of regulations, true positives with protein-protein interaction data. $\hat{TP}$ are edges predicted with incorrect direction and k -skip validation is using gene cascades of k genes.

#genes	model(o)	# edges at delays		#Edges	Validation			likelihood
		1	2		TP	$\hat{TP}$	k -skip TP	
Chou(9)	d(1)	17		17	4	5	9	[-139.39,-137.9]
	h(2)	10	16	26	7	7	20	[-130.61,-129.38]
Mtb(9)	d(1)	12		12	4	1	5	[-68.46,-66.42]
	h(2)	17	14	31	7	4	16	[-59.07,-58.50]
Hela(51)	d(1)	159		159	13	6	19	[-2085.78,-2079.16]
	h(2)	102	144	246	14	4	61	[-2104.92,-2092.19]

3.4.5.3 *M.tuberculosis* DNA Repair Data

The first-order DBN could only recover 6 TP in the MTB pathway. True positives include activation of *lexA* by *linB* which is known to initiate the dna-repair pathway. Regulation of *ruvC* by *fadD23* which appears at the end of the repair also was correctly identified. However, DBN was unable to identify several time-delayed interactions. Hence we resort to a second-order VDBN on the data. This added another 8 TP to the prediction. 14 of the 31 edges predicted were second order indicating that delays are significant for the pathway. This is in accordance with the fact that the bacteria replicates very slowly. The new true positives were regulation of *ruvC* by *fadD23*, and activation of *ruvC* by *lexA*. *RuvC* is known to be part of an alternate dna repair mechanism which is probably activated with delays. The binding of *recA* with *dnaB* which starts the pathway was also correctly predicted. Second-order TP were binding of *lexA* to *dnaB* and *recA* activating *dnaE2* which are critical to the pathway. Likelihood of VDBN is slightly higher than DBN showing a better fit to the data. k -skip validation looking at cascades of two genes could recover all 16 edges in the pathway. Confidence of hdbn edges (Fig. 3.3) is much higher than the dbn interactions.

Figure 3.2: Time-delayed interactions in a predicted network using HDBN bootstrap in yeast cell cycle.

3.4.5.4 Human Hela Cell Cycle

The importance of time-delays becomes significant in regulatory networks, especially when a large number of genes are involved. In general, the first-order interactions are presumed to be instantaneous events such as initialization, silencing, etc.; the time-delayed interactions were mostly regulatory or feedback associated. The DBN could only predict 19 interactions correctly in the pathway suggesting more time-delayed edges. Next, we fitted a second-order VDBN to the time series data. Though likelihood of the graph did not change significantly over 144 edges were assigned second-order by the model. In Fig. 3.4 several new interactions are identified by the VDBN for the Hela cell cycle. Since time-delayed information is not easily available for such a large network we looked at the top nine hubs conserved in the pathway. We also did a k -skip validation where cascades of two genes were considered for second order delay. k -skip validation saw a significant increase in true positives to 61.

The top hubs predicted by DBN and HDBN are compared with Biogrid in Table 4.7. The DBN identified hubs FGF1, FGF18 and C-Jun correctly. We looked at the function of these hubs and found that indeed they were

Figure 3.3: Time-delayed interactions in a predicted network using HDBN bootstrap in *M.tuberculosis* DNA repair pathway.

instantaneous such as initiation of cell division by FGF or involved in apoptosis such as PUMA. C-Jun levels are constant in the cell-cycle indicating constant feedback and regulation. The VDBN had GADD45 as a hub which is known to be mediated by the hub P53 given by BioGrid. Cyclin1 and Bcl-X1 are also known to co-operate in induction of cell cycle proliferation. IAP which is a inhibitor of apoptosis is predicted by VDBN as it has a regulatory function involving delays. Thus, we are convinced that the VDBN outperforms the DBN in identifying functionally important genes and hubs.

3.5 Discussion

GRN are extremely complex where different genes interact with different delays and topological arrangements hence time-series gene expression analysis is more suitable for this problem. Though several authors have looked at

Figure 3.4: Time-delayed interactions in a predicted network using HDBN bootstrap in Hela human cell cycle.

Table 3.7: Top 9 hubs obtained for 51 genes in Hela cell line of human cell-cycle, ranked in the order of their connectivity. The hubs obtained from the BioGrid are also given for comparison.

model(σ)	Rank of genes based on connectivity								
	1	2	3	4	5	6	7	8	9
BioGrid	FGF1	P53	FGF18	PUMA	BCL2	NOXA	BCL-XL	C-JUN	P21
d(1)	PUMA	FGF1	FGF18	GADD45	CYCLINE1	C-JUN	RNAHP	A20	TRF4-2
h(2)	PUMA	FGF18	CYCLINE1	PAI	MCL1	C-JUN	NFKB	BCL-XL	IAP

gene regulatory network prediction, time-delayed interactions and the effect of number of available time points on network size has not been studied.

While DBN can consider single time point delays it is unable to predict longer delays in regulation. The HDBN is able to overcome this limitations as it can predict interactions dependent over two, three or four delays. Simulation on real datasets shows that HDBN is also able to predict new interactions which are not seen by the DBN. Confidence of edges in HDBN is considerably higher than the DBN and so is the likelihood of the predicted network.

Using the HDBN comes at a cost of increasing the parameters of the model. The number of time points needed to predict accurately also increases with the order of the model. Hence, we opted for a reversible jump MCMC where optimal delay is determined by increasing or decreasing a single delay. While

HDBN is able to recover network with few time points when the number of genes is low, the prediction is poor as the number of genes becomes large.

Synthetic data demonstrates the stability of HDBN when sufficient time points are available, however time points are often limited in microarray experiments. Bootstrapping can help to identify false positives. A comparison with vector sieve bootstrap for continuous data shows that the multinomial bootstrap requires much fewer time samples to estimate error distributions. Investigation on real data reinforce our inference using synthetic data.

Chapter 4

Building GRN with Skip-chain Models

Time delays have an important role in gene regulatory events. Biological networks feature varied coupling of slow and fast activities [94]. Learning algorithms for higher-order Bayesian networks compute the maximum likelihood (ML) parameters for each time delayed state. The number of parameters become intractable when the delays or orders of the network are large. In this chapter we suggest use of a skip-chain Markov model to handle long delays in regulatory networks. Storing of information for long periods of time is done using latch nodes which are independent of other nodes in the network [95]. When a skip-chain model is unfolded in time the long delays appear as jump ahead connections which provide a shorter path for propagation of probability [96]. Since delays are cascaded together they don't need to pass through non-linearity at each time point as in a VDBN.

We first explain our approach consisting of two stages namely identification of candidate interactions for long delays and using Viterbi forward algorithm to compute the ML parameters for them. Secondly, we discuss a Dirichlet conjugate prior which has been previously applied to skip-edges in sequences [97]. Lastly, we validate the model using synthetic as well as real protein interaction networks and construct confidence indices similar to that for VDBN across probabilistic residual bootstrap samples. In essence we show how our model is able to outperform previous prediction models for gene regulatory networks from time-series expression data.

The DNA repair system of *Mycobacterium tuberculosis* typically spans 7 hrs and is unusually slow compared to other genomes making it expensive to capture in time series experiments. We propose to study this phenomena in gene networks using our skip-chain model. We also look at benchmark differentially expressed genes in cell division cycle in yeast and humans of significance

in cancer and spanning around 2 hrs. Several time-delayed interactions were observed with our analysis. The presence of hubs in the networks indicates that a small number of transcription factors regulate the rest of the genes and maintain robustness of cell signaling in the face of perturbations. Skip-chain models are capable of conveniently capturing long-distant and time-delayed regulations in such architectures.

We follow the outline of VDBN model starting with a description of a skip-chain model state diagram where a very long delay can be computed as a Viterbi forward probability over a cascade of intermediate genes. A prior over skip-edge parameters is investigated so that we can avoid over-fitting of model to the data. The residual probabilistic bootstrap can generate new samples from short time-series and assess the confidence of predicted edges. We conclude the chapter with stability between the delays and validation with functional knowledge of the pathways.

4.1 Motivation and Contributions

Fluctuations in gene expression levels are commonly seen in biological processes such as transcription and translation resulting in time delays. Such delays could trigger variable response pathways and hence affect the stability of the cellular system. Historical attempts to treat the long-delay problem include non-linear differential models operating on multiple time-scales [98], Markov systems with jumping models between delays [99], [100], Bayesian clustering algorithms to couple genes and also change point algorithms to determine time-delayed variations in genes [101]. The fraction of probability in state variables diminishes rapidly with delay, hence a heavier weight fraction for information more into the past was in used in [102].

Cellular activities of gene and gene products represented in gene regulatory networks (GRN) provide a basis for signal transduction pathways. Since, the signal networks are often presumed to be stable, meaning that slight changes in the state of a few parents do not change the expression state of the child gene. The phenomena relates to the redundancy of biological systems which are to ensure that the system retains functioning in spite of perturbations. Thus the networks follow a distributed architecture with a few hubs regulating longer cascades of gene pathways well represented in skip-edge flow model.

Internal thermal noise as well as external noise due to random variations in external control parameters of the lab experiment results in distorted gene expression measurements [103]. Perturbations could include random failures,

repairs of components, and sudden environmental changes. The problem is amplified in larger networks. We deal with noise in long delays by imposing a Dirichlet conjugate prior dependant on the shorter delay parameters. Bootstrapping is also used to assess the confidence of edges and delays in the presence of noise.

The dynamic Bayesian networks (DBN) can model the temporal dynamics where the parents from the previous time instant are assumed to be regulating the genes [60]. This first-order assumption allows feedbacks but still deprives DBN of representing time-delayed interactions. The DBN when extended to higher-order is capable of representing delayed interactions [64]. However, these generative models become computationally intractable at very high-orders. Therefore, we propose a skip model which can handle long term delays in regulatory interactions [13], [36]. The skip model is represented by two types of features: (a) linear features modeling the short-delays; and (b). skip features modeling long-delays. In our model, skip features are modeled by using a hidden Markov model (HMM) where the log likelihood of the network is decomposed into a sum of consecutive pairs of genes. So the maximum likelihood estimate of regulation can be found by using the Viterbi algorithm(VA).

The linear feature attempts to model interactions which occur instantly or with little delay. The skip feature model interactions occurring much later in the pathway, for example, a gene i inhibits a gene j to start a process, and later gene i regulates another gene k towards the end of the process. The skip-feature probability is decomposed into a sum of terms for consecutive pairs of genes in the time-course and the most likely interactions are found using the Viterbi algorithm [104]. The Viterbi skip-feature can automatically determine the best time delay in a higher-order Markov chain representing the instantaneous network of DBN [105], [97].

Our approach consists of two stages: (a) identification of time-delayed interaction features and computation of Viterbi scores; and (b) prediction of the optimal GRN by using a reversible MCMC. The acceptance mechanism of the MCMC includes the Viterbi scores of the skip model. Use of priors on shorter time series is done to address over-fitting. We demonstrate our method with an application to a long time-series of yeast cell-cycle data. We find core genes that are earlier known to have regulatory effects with differing time-delays on the cell cycle. Experiments were performed on larger sets of genes and higher number of time points obtained on cell cycle regulation.

Further, we show validation of our results by employing protein-protein interaction (PPI) data. DNA in a cell contains genes which are converted

to mRNA (expressed genes) through transcription and then translated into proteins. Complex formation and other critical interactions that regulate the biological processes happen at protein levels. Hence validation using protein interaction networks is a reliable way of assessing accuracy of our predicted model. A validation with BioGrid PPI [34] for higher-order interactions shows that the method is more effective than existing techniques. We also used skip models to find time-delayed interactions in *M.tuberculosis* [106]. In this chapter, we have detailed our formulation on skip models and describe implementation on a reversible MCMC.

4.2 Parameter Estimation for Skip-Chain Model

Skip-chain model can be viewed as a first-order hidden Markov model (HMM) connecting different layers of HDBN networks. MAP estimate can be used to determine the best time-delayed interaction between genes using Viterbi forward path through these layers. We can extend the equation for HDBN given by (3.2) to include the MAP estimate as follows

$$p(x|s, \theta) = \prod_{i=1}^n p(x_i^t | a_i^t, \theta_i^t) \prod_{r'=1}^r \prod_{i=1}^n p(x_i^t | a_i^{t-r'}, \theta_i^{t-r',t}) \arg \max_{y_{s_t:t}} p(x | y_{s_t:t}) p(y_{s_t:t}) \quad (4.1)$$

where $x = (x_1, x_2, \dots, x_n)$ denotes the vector of gene expressions of n genes, $p(x_i^t | a_i^{t-r'}, \theta_i^{t-r',t})$ denotes additional parameters for the conditional probability of gene expression x_i given its parent gene expressions a_i from previous r time points, $\theta_i^{t-r,t}$ denotes the parameters of the conditional probabilities of the transition networks and the last term in (4.1) is the MAP estimate of path from previous time instant s_t to t . First-order assumption requires only a $n \times n$ transition matrix to compute even very long delays (up-to order $n - 2$) with few time samples and is computationally much faster.

4.2.1 Hidden Markov Models (HMM)

The classical DBN is unable to capture complex time-dependencies and is extended to a r -order ($r \geq 2$) Markov chain. It predicts the expression levels of a set of genes based on the expressions of up to r previous time points using frequency statistics. Higher-order dynamic Bayesian networks (HDBN) have been proposed to study time-delayed interactions. However, as the order o

increases it is not possible to predict the delays because of the difficulty in estimating the increasing number of parameters.

Therefore, we resort to a first-order hidden Markov model (HMM) to determine delayed interactions. It determines the probability of expression of a gene j at time point t , given that gene i was observed at s_t where $s_t < t - 1$ within the section of the time-series of length $t - s_t$. Let a sequence of hidden states from time point s_t to t be denoted by $y_{s_t:t} = y(t')_{t=s_t}^t$ where $y(t')$ denotes the gene expressed at time point t' in the path. Correspondingly, we have the observed data $x_{s_t:t} = x(t')_{t=s_t}^t$ where $x_i(t') = k$ is the discretized gene expression state.

Given the microarray data, the maximum likelihood estimation can be used to estimate the state transition and emission probabilities, which are defined as follows [32]:

$$a_{l,m} = \frac{M_{l,m}}{\sum_{m'=1}^n M_{l,m'}}, \quad \forall y_{t'} = g_l, y_{t'+1} = g_m \in G \quad (4.2)$$

$$b_l(k) = \frac{M_l^k}{\sum_{k'=1}^d M_l^{k'}}, \quad \forall k \in \Gamma, t' \in \{s_t, s_t + 1, \dots, t\}, g_l \in G. \quad (4.3)$$

where $M_{l,m}$ denotes the number of occurrences where $x_{l,t'} = x_{m,t'+1} = 1 \quad \forall t' \in \{s_t, s_t + 1, \dots, t\}$, $\forall k \in \Gamma$ and M_l^k denotes the number of occurrences where gene g_l has been at discrete state level k , $\forall t \in \{s_t, s_t + 1, \dots, t\}$.

4.2.1.1 Viterbi Algorithm

When the expression time-series are modeled with an HMM, the maximum a posteriori (MAP) estimate could be used to find the time-delayed interactions of a pair of genes. The path begins and ends at the known states of genes : say, $y_{s_t} = g_i$ and $y_t = g_j$. We assume that $t - s_t$ is not very large and conditional independence between feature vectors. For a sequence of a set of genes, the most probable path is given by the MAP estimate:

$$\arg \max_{y_{s_t:t}} p(y_{s_t:t} | x) = \arg \max_{y_{s_t:t}} p(x | y_{s_t:t}) p(y_{s_t:t}) \quad (4.4)$$

VA can be used to find the best path by finding the MAP estimate, between two genes at distant time points. It is a dynamic programming procedure and determines the best path in an incremental manner. Let $\delta_m(t')$ be the probability of the most probable path ending at gene g_m with the observation $x_{m,t'}$ at time t' . Then, the best path at the next iteration is found as

$$\delta_m(t' + 1) = b_m(k) \max_l \{ \delta_l(t') a_{l,m} \} \quad (4.5)$$

We can divide the path probability by length of path to get a first-order probability as a goodness of fit of the path. Hence the skip-edge score is the normalized MAP interaction :

$$h(x_i, a_i, s_t, t) = \log \frac{1}{(t - s_t)} \max_{y_{s_t:t}} p(y_{s_t:t}|x) \quad (4.6)$$

where the parent set a_i has only one gene at time point s_t .

Finally, for any pair of genes, the best time-delayed interaction has the highest probability :

$$\hat{h}(x_i, a_i, s_t, t) = \max_{(s_t, t), t-s_t > 0} h(x_i, a_i, s_t, t) \quad (4.7)$$

where $g_j \in a_i$ and r is predefined linear order. We find the corresponding delay if $\hat{h}(x_i, a_i, s_t, t)$ is greater than a predetermined threshold. Fig. 4.1 illustrates a Viterbi forward path as a bold line in the time-series data. The lower order delays of the VDBN are shown as dotted lines .

Figure 4.1: State transition diagram illustrating six time points and four genes in a DBN. The states $\{-1, 0, 1\}$ represent down-, un-, and up-regulated genes. The dashed edges are linear $r = 1, 2$ order edges found by linear features. The solid edge represents a skip-edge over four time points.

4.2.1.2 Linear and Skip features

In order to handle both short- and long-range interactions, we model gene regulations by using a linear model and a skip model. The skip -chain model is illustrated in Fig. 4.2. The skip-edge appears as a jump ahead connection in the unfolded network over time providing a shorter path for propagation of probability.

The two types of features are expressed in the likelihood of a gene expression x_i as a weighted sum of linear and skip-edge scores. For gene g_i :

$$\log p(x_i|a_i, \theta_i) \propto \lambda f(x_i, a_{i(t-o:t)}, t) + (1 - \lambda)h(x_i, a_i, s_t, t) \quad (4.8)$$

where $f(x_i, a_{i(t-o:t)}, t)$ and $h(x_i, a_i, s_t, t)$ represent the linear- and skip-feature functions and λ is the interpolation weight.

Linear-chain feature functions $f(x_i, a_{i(t-r:t)}, t)$ represent local dependencies that are consistent with an r -order Markov assumption of gene expressions. The skip-chain feature functions $h(x_i, a_i, s_t, t)$ exploit the dependencies between genes that are arbitrarily distant at time instances s_t and t respectively [13]. It can model variable length Markov chain up to $T - 1$ order. We use DBN to implement a linear-chain model and first-order HMM to implement the skip-chain model.

The optimal delays can be found by using a GA to optimize the likelihood. For an r -order HDBN, we can have $o^{|a_i|}$ structural possibilities for each gene where $|a_i|$ is cardinality of parent set. Hence, the search space and corresponding complexity is very high to find delays. On the other hand, skip models use VA to find the optimal delay and associated probability whose complexity is only quadratic on length of delay [32].

4.2.2 Dirichlet Prior for Skip-Edge

Most higher-order Markov models are far from optimal. They are extremely sensitive to change in pathways and associated data. This happens as most of the data is general rather than feature specific for an interaction. The goal of adaption has been to make good use of available feature data and reduce the over-fitting in the model. Our adaption model combined the reliable general DBN with a volatile feature specific HMM for long delays. We further extend the MLE to a Bayesian learning where a Dirichlet conjugate prior is imposed on each of the parameters.

Given the set of conditional distributions with parameters $\theta = \{\theta_i : i = 1, 2, \dots, n\}$, the likelihood can be written as

$$p(x) = \int p(x|S, \theta)p(\theta|S)d\theta \quad (4.9)$$

The integral can be easily written in a closed form due to conjugacy between Dirichlet and multinomial distribution. However, we can alternatively maximize probability as (MAP) :

$$\theta_{MAP} = \arg \max_{\theta} p(x|\theta)p(\theta) \quad (4.10)$$

Using the linear feature as a Dirichlet conjugate prior [97] for the skip feature of a gene we get :

$$\theta_{MAP} = \arg \max_{\theta} \prod_{i=1}^n \theta_i^{h'(x_i)h(x_i) + \alpha f(x_i) - 1} \quad (4.11)$$

where $h'(x_i) = \sum h(x_i)$

where $h'(x_i)$ is total probability of the skip-path, α is a weighting factor between the linear and skip features.

Next, we can specify the interpolated probability of gene g_i based on linear and skip-edges.

$$\theta_i = \lambda f(x_i) + (1 - \lambda)h(x_i) \text{ where } \lambda = \frac{\alpha}{\alpha + h'(x_i)} \quad (4.12)$$

here, instead of using a constant, λ is estimated using prior linear feature and the total probability of the skip path.

Figure 4.2: A skip chain model has overlapping skip-edges which model long-distant dependencies.

4.3 Experiments with Synthetic Data

Scale-free network topologies were generated with in-degree d_i set to a maximum of 5 genes and $\gamma = 1$. Since GRN are causal we randomly flipped the direction of edges to create feedback loops while generating the network [75]. Delays of order two were randomly added to the edges. GRNs defined by scale-free networks of 10 genes were simulated and the parameters, or the conditional probability distribution (CPD) tables $\{\theta_{ijk}\}_{i=1, j=1, k=1}^{I, J, K}$, were fixed for all bootstrapped samples. The genes were assumed to take three states $\{-1, 0, +1\}$ representing down-, no-, and up-regulation of genes. Genes with

no parents were set to up-regulation with corresponding $CPD = \{0.1, 0.1, 0.8\}$; for all remaining genes the state was determined by the dominant state in the parent set.

4.3.1 Structure Learning

Simulations were performed to evaluate VMCMC in finding the optimal structure of skip-chain DBN. Initialization of the structures was done using mutual information. The histograms of mutual information were plotted for each pair of genes. The peaks of the histogram were taken to determine the threshold for the presence of regulatory connections.

MCMC was used for structural learning with 50000 iterations for convergence for networks of size 10, 20 and 100 and delays of four to seven time points. A Dirichlet prior was used as in the case of simple DBN.

Performance measures such as precision¹, recall², and F-measure³ were evaluated using true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), respectively.

4.3.2 Effects of the Number of Time points

Given the size of the network being built, in order to study the effects of the number of time points, gene expression values were generated up to a time-span $T \in \{50, 100, 200\}$ time points from GRN, assuming skip-chain DBN as described in Section 4.2. First, gene expression at the first time point were initialized randomly; second, gene expressions at other time points were sampled from a skip-chain model, using (4.8); and third, Gaussian noise with variance $\sigma^2 = 0.1$ were added. Structural learning was done using VMCMC simulations. The convergence was determined when there was no change in fitness or generations reached 50000 iterations. For a given I number of genes and T number of time points, $B = 100$ bootstrap time-series dataset were generated. We compared the use of Dirichlet conjugate prior for skip-edges on a network of size 100 and up-to five time delays when 50 time samples are used. It can be seen that using a prior gave fewer FP and FN. The recall of the network was much higher at 0.86 when a prior is used compared to only 0.58 when no-prior is used.

Table 4.2 shows convergence and stability of building networks of different sizes with a varying number of time points and for delays four to seven. As

¹Precision = $TP/(TP + FP)$

²Recall = $TP/(TP + FN)$

³F-measure = $2 \frac{Precision \times Recall}{Precision + Recall}$

Table 4.1: Accuracy of networks derived using prior for skip-edges and no-prior when five time delays are considered on 100 genes and 50 time points

Method	FP	FN	Precision	Recall	F-measure
Prior	178	55	0.53	0.86	0.67
No-prior	131	25	0.43	0.58	0.48

seen, to obtain decent accuracy and stability, the number of time points needed should approximately be equal to the number of genes in the network. Interestingly, our simulations indicate that small networks required more time points, relative to the number of nodes in the networks, than the large networks. For shorter time-series, precision and recall measures were low for long-delays, indicating that the performance was adversely affected by the presence of large number of false negatives (FN) and false positives (FP) (more by FP). Increase in the number of time points improved the overall performance; Note that the improvement was higher for 20 gene network compared to 10 genes however a decrease was seen with 100 genes . It is easier to correct false negatives by increasing the number of time points than the false positives.

It can be seen that FP are higher for longer delays, for example a, 100 genes network has 13 order 4 FP when simulated with 50 time points. This suggests that longer delays are more difficult to predict in the skip-model compared to smaller delays. Similar to chapter 1 the accuracy of the skip-model decreases with increasing order and lower number of time points. The stability of the network is almost half when using 50 time points compared to 100 time points. Accuracy of 100 gene networks is much lower in higher orders though networks are stable.

4.3.3 Stability of the Networks

The last column of Table 4.2 shows that the stability of learning GRN structure by using skip-chain DBN decreases with the number of time points relative to the network size. For 20 genes an MCMC has remarkably high stability while for 100 genes both methods had poor stability.

The stability between delays of length one, two or three is shown in Table 4.3 . The pattern is similar for a smaller network of 10 genes and bigger network of 100 genes where longer delays have higher stability. However, the network with 20 genes shows a decline in stability for order six delays. This can be related to the results that bigger networks converge faster than smaller ones. Zero delay stability implies the edge has the same delay in both networks being compared and is usually low.

Table 4.2: Accuracy of building skip-chain MCMC model with Dynamic Bayesian Networks in the presence of 10% Gaussian noise.

#Genes	#Time	Method	FP(1,2,3,4,5,6,7)	FN(1,2,3,4,5,6,7)	Precision	Recall	F-measure	Stability
10	4	50	2(0,1,1,0)	1(0,0,1,0)	0.99	0.99	0.94	0.87
		100	2(1,1,0,0)	0	0.99	0.99	0.94	0.87
	5	50	11(2,3,1,2,3)	4(0,0,2,0,2)	0.6	0.78	0.66	0.71
		100	2(0,0,1,0,1)	0	0.9	0.99	0.95	0.87
	6	50	11(1,2,2,2,2,2)	1(0,1,0,0,0,0)	0.61	0.95	0.74	0.56
		100	2(0,1,0,0,1,0)	0	0.91	0.99	0.95	0.87
	7	50	14(2,2,2,2,2,2,2)	1(0,0,0,1,0,0,0)	0.56	0.96	0.71	0.54
		100	4(1,0,1,1,1,0,0)	0	0.83	1	0.9	0.81
	20	50	27(7,5,5,10)	3(0,0,2,1)	0.56	0.93	0.7	0.52
		100	6(1,1,1,3)	1(0,0,1,0)	0.87	0.97	0.92	0.82
		50	35(6,7,6,7,9)	3(2,1,0,0,0)	0.51	0.93	0.65	0.48
		100	9(2,2,2,1,2)	1(1,0,0,0,0)	0.81	0.96	0.88	0.76
		50	42(8,6,6,7,7,8)	2(0,0,1,0,0,0)	0.46	0.95	0.62	0.44
		100	10(2,1,2,1,2,2)	1(0,0,1,0,0,0)	0.79	0.9	0.8	0.76
		50	53(5,9,8,8,7,7,9)	2(0,1,0,0,1,0)	0.41	0.95	0.57	0.39
		100	11(2,2,1,2,2,1,2)	1(0,0,1,0,0,0,0)	0.77	0.99	0.87	0.75
100	4	100	35(5,9,7,13)	10(2,7,0,1)	0.85	0.95	0.89	0.8
		200	1(1,0,0,0)	1(1,0,0,0)	0.99	0.9	0.99	0.89
	5	100	44(9,8,8,7,12)	10(1,0,4,4,1)	0.81	0.95	0.87	0.76
		200	1(0,0,0,0,1,0)	5(0,0,3,2,0)	0.99	0.97	0.98	0.93
	6	100	53(7,9,10,8,9,10)	4(0,1,2,0,0,1)	0.78	0.98	0.87	0.73
		200	1(0,0,0,0,0,1,0)	1(0,0,1,0,0,0,0)	0.99	0.99	0.99	0.94
	7	100	266(35,37,39,38,33,35,49)	1(0,0,0,0,0,1,0)	0.42	0.99	0.59	0.4
		200	23(3,3,3,3,3,3,5)	0	0.87	1	0.99	0.82

4.3.4 VAR Networks

Table 4.4 shows a comparison of VAR with HDBN model for network of size 10 and 100 time points. The VAR with delays on the 10 gene network performs very poorly confirming the non-linear nature of delayed pathways. GBN performance is much higher than VAR however compared to skip-chain model it performs poorly when data is generated from the model.

4.3.5 Vector Sieve Bootstrap

To bootstrap we used residual bootstrapping to generate more samples. A comparison was done of probabilistic residual bootstrapping and vector sieve bootstrap where noise was added from error distribution. For vector sieve we used a 10 gene set with delays up to order five and 50 time points 500 bootstraps were resampled. It was not possible to do residual bootstrap for long delays using VAR model. Table 4.5 shows that probabilistic bootstrapping at a threshold of 0.5 predicts 2 fewer FP than the original time series and hence can correctly assess confidence of edges. We can hence conclude that probabilistic bootstrapping is optimal for GRN prediction. When CPD are specified using 100 time points. The bootstrap method can recover the network perfectly and has previously been shown in Table 4.5.

Table 4.3: Stability between delays of length zero, one, two and three when building HDBN-MCMC model with Dynamic Bayesian Networks in the presence of 10% Gaussian noise.

#Genes	Order	#Time	1	2-3	4-5	6-7
10	4	50	0.36	0.45		
		100	0.36	0.44		
	5	50	0.18	0.36	0.35	
		100	0.18	0.39	0.33	
	6	50	0.19	0.26	0.25	
		100	0.23	0.31	0.31	
	7	50	0.11	0.22	0.23	0.25
		100	0.13	0.26	0.26	0.27
20	4	50	0.33	0.42		
		100	0.36	0.45		
	5	50	0.16	0.33	0.33	
		100	0.19	0.38	0.38	
	6	50	0.2	0.28	0.25	
		100	0.23	0.28	0.25	
	7	50	0.1	0.19	0.21	0.25
		100	0.13	0.24	0.27	0.32
100	4	100	0.36	0.44		
		200	0.34	0.42		
	5	100	0.19	0.37	0.4	
		200	0.19	0.37	0.39	
	6	100	0.23	0.31	0.29	
		200	0.24	0.32	0.3	
	7	100	0.12	0.24	0.25	0.29
		200	0.13	0.25	0.26	0.3

Table 4.4: Accuracy of networks derived using VAR and HDBN when five time delays are considered on 10 genes and 100 time points

Method	FP	FN	Precision	Recall	F-measure
VAR	25	5	0.23	0.58	0.32
GBN	10	10	0.44	0.44	0.44
Skip-chain	0	0	1	1	1

4.3.5.1 Bootstrap on Real data

We simulated the three datasets described in Chapter 1. Preprocessing and discretization were also performed. Here we use uninformative priors $\alpha_{ij} = q_i$ and $\alpha = 20$. We show validation of our results by employing BioGrid protein-protein interaction (PPI) data [34] marked as TP and edges with wrong directions marked as $\hat{TP}$. Table 4.6 shows the bootstrap confidence intervals for the three datasets as well as validation using protein interaction data. Some edges are identified with wrong directions shown as $\hat{TP}$ in the graphs. A k-skip validation for time-delayed interactions is also performed. It can be seen that the skip-chain DBN has a higher likelihood than corresponding HDBN and also much higher true positives.

We generated 100 bootstrap samples using probabilistic residual bootstrap and 100 different starting structures using mutual information threshold of 0.1. Fig. 4.3 shows the error distributions in the four datasets. It can be seen that real networks have variable error probabilities suggesting the significance of the residual probabilistic bootstrap.

Table 4.5: Accuracy of residual bootstrap for 50 time points and up-to five time point delays

#Model	Method	FP	FN	Precision	Recall	F-measure
HDBN	Original	6(1, 1, 2, 1, 1)	2(0, 0, 2, 0, 0)	0.72	0.88	0.79
	Bootstrap	4(1, 0, 1, 1, 1))	5(0, 1, 2, 0, 2)	0.76	0.72	0.74
	Bootstrap(200)	0	0	1	1	1

Figure 4.3: Distribution of error probabilities of predicted edges of regulatory networks obtained using VMCMC(a) Chou *et. al.* (9 genes) having up-to five delays (b) Tuberculosis (9 genes) having up-to four time point delays (c) Hela (51 genes) having up-to seven time point delays.

4.3.5.2 Yeast Cell-Cycle Data

Since the VDBN is unable to identify long delayed interactions across G1 phase of cell-cycle, we model the data using a fifth order skip-chain DBN. It gave 16 TP which was higher than those by VDBN. However direction of causality was incorrect for 9 TP. The likelihood of the network showed a remarkable increase compared to the linear DBN and VDBN models. An example of a fifth order TP was regulation of Ndd1 by Swi4 spanning over 40 mins in the real cell cycle. Also it identified Ace2 regulating Cln3 at five time points delay and can be validated in the benchmark network. It can be seen that the skip-chain predicted over 8 interactions with moderate and 18 edges having very long delays confirming the occurrence of higher-order interactions. Fig. 4.4 shows several time-delayed interactions predicted by the skip-chain model. Bootstrapping has been done to assess confidence of edges and reduce the false positives. Several TP have higher confidence compared to the false positives. Dirichlet prior with pseudo count 20 was used for the yeast network.

4.3.5.3 *M.tuberculosis* DNA Repair Data

We used a fourth-order skip-chain to model the time-delayed interactions in tuberculosis DNA repair. Surprisingly 32 of the predicted 40 edges had a fourth delay interaction corresponding to the slow nature of the pathway. We could

Table 4.6: First- and higher-order regulations predicted by DBN(d), HDBN(h), and skip-chain(d:h) models built on datasets by Chou *et. al.* (9 genes), Tuberculosis (9 genes) and Hela cell line(51 genes). The number of genes in the dataset, (likelihood), total number of regulations, true positives with protein-protein interaction data. $\hat{TP}$ are edges predicted with incorrect direction and k -skip validation is using gene cascades of k genes.

#genes	model(<i>o</i>)	# connections at delays						#Edges	Validation			likelihood
		1	2	3	4	5	6		TP	$\hat{TP}$	k -skip TP	
Chou(9)	d(1)	17						17	4	5	9	[-139.39,-137.9]
	h(2)	10	16					26	7	7	20	[-130.61,-129.38]
	d:h(1:5)	5	4	4	6	12		31	7	9	20	[-100.68,-99.92]
Mtb(9)	d(1)	12						12	4	1	5	[-68.46,-66.42]
	h(2)	17	14					31	7	4	16	[-59.07,-58.50]
	d:h(1:4)	1	1	6	32			40	8	8	16	[-36.74,-36.37]
Hela(51)	d(1)	159						159	13	6	19	[-2085.78,-2079.16]
	h(2)	102	144					246	14	4	61	[-2104.92,-2092.19]
	d:h(1:6)	102	144	31	62	56	67	462	20	10	81	[-1783.01,-1777.77]

also correctly predict all the 16 edges reported in the benchmark method. The likelihood of the network was much higher than DBN and VDBN. Most of the true positives were assigned fourth order delay. Confidence of TP was high while confidence of wrong edges was lower. There were 26 FP in the system, however this could be due to lack of biological evidence for those interactions. Most of the false positives involved gene Rv2719c which is still hypothetical and the dnaE2 gene. Confidence of skip edges (Fig. 4.5) is much higher than the hdbn interactions.

4.3.5.4 Human Hela Cell Cycle

The VDBN model could identify 61 of the 81 edges in our validation set from BioGrid and KEGG. We further applied our skip-chain model with order six delays to the time series data. The model correctly identified 30 edges and had a much higher likelihood compared to the DBN and VDBN model. Several interactions were assigned moderate as well as long delays upto six time points emphasizing the time-delayed dynamics in a human cell cycle. A k -skip validation could recover all the 81 edges in our validation set. Interestingly the hubs of the skip-chain model were a combination of those predicted by the DBN such as A20 and C-Jun and also those by VDBN such as NFkB. Hence, we can conclude that the model is able to detect short as well as long delays accurately. Particularly the skip-chain identified Noxa as a hub which regulates intracellular susceptibility of cells to apoptosis.

Figure 4.4: Time-delayed interactions in a predicted network using skip-chain DBN bootstrap in yeast cell cycle.

Table 4.7: Top 9 hubs obtained for 51 genes in Hela cell line of human cell-cycle, ranked in the order of their connectivity. The hubs obtained from the BioGrid are also given for comparison.

m(o)	Rank of genes based on connectivity								
	1	2	3	4	5	6	7	8	9
BG	FGF1	P53	FGF18	PUMA	BCL2	NOXA	BCL-XL	C-JUN	P21
d(1)	9	9	8	5	5	5	5	4	4
	PUMA	FGF1	FGF18	GADD45	CYCLINE1	C-JUN	RNAHP	A20	TRF4-2
	17	12	11	11	10	9	8	8	8
h(2)	PUMA	FGF18	CYCLINE1	PAI	MCL1	C-JUN	NFKB	BCL-XL	IAP
	30	18	14	14	13	12	12	12	11
d:h(1:6)	GADD45	RG55	C-JUN	NOXA	A20	NFKB	CYCLIND1	PAI	IKAPPABA
	28	25	23	23	23	23	22	22	22

4.4 Discussion

Pathways are often triggered by transcription factors which in turn express genes and produce proteins. Therefore, the regulatory interactions in molecular pathways can be given by GRN. Gene regulation generally includes dynamic feedback loops, cascaded interactions, intermediary factors, etc., which provides for underlying biological mechanisms of regulation. This results in different time delays in regulatory interactions.

We have considered higher-order DBN (HDBN) for representing delays in regulations. When larger delays are involved, implementation of HDBN be-

Figure 4.5: Time-delayed interactions in a predicted network using skip-chain DBN bootstrap in *M.tuberculosis* DNA repair pathway.

comes intractable. Therefore, we proposed a skip-chain HDBN. This involved two components: linear model to represent short delays and skip model to represent long delays. These two components may represent actions of activator and inhibitor involved in regulatory interactions. Our method was evaluated against earlier approaches, which shows our method better fits the gene expression data when GRN was built. In order to provide a more biological meaningful validation, we performed comparison with the protein-protein interaction data. That validation also showed the superiority of our technique over other methods but because of the incompleteness of PPI data sources, such comparisons result in large false positives.

Skip-chain models address the difficulties of a HDBN by easily incorporating long time-delayed regulations. The skip model is a first-order HMM and captures long-distance dependencies of input time-course gene expressions. This inference technique leads to lower computational times without loss in the accuracy compared to HDBN. The forward Viterbi path through the trellis determines the best long-distant time delay and therefore automatically finds the best higher-order interactions between genes. The method gives

Figure 4.6: Time-delayed interactions in a predicted network using skip-chain DBN bootstrap in Hela human cell cycle.

more accurate and biologically plausible networks than DBN implementations as short- and long- time delays are inherent in biological interactions.

An organism responds to changes in its environment by altering the level of expression of critical genes. The virulence of *Mycobacterium tuberculosis* depends on the ability of the bacilli to switch between replicative (growth) and non-replicative (dormancy) states in response to host immunity. Different transcription networks are active at different stages of the response. The coordinated repression of genes is likely to contribute to survival by conserving energy and precursors under nutrient-limiting conditions and/or minimizing expression of potential antigens.

M. tuberculosis is known to have an unusually long period of 10 hrs for the DNA replication fork to traverse the chromosome. Our results showed several interactions at 4 hrs in the *M.tuberculosis* DNA repair pathway. An order-4 HDBN with skip-chain dependencies was shown to outperform ordinary HDBN's. For genes to interact they both have to be upregulated. We use this property to select events where a pair of genes are both upregulated at similar or delayed time points. It is well established that interacting genes have correlated expression patterns. To this end, we add the interactions at non-consecutive time points. This is because a DBN assumes a first-order network

and is not able to model complex time-delayed interactions. We assumed that all interactions had equal priors. However our method is able to distinguish between short- and long-term interactions and hence allow us to make a better judgement on *M.tuberculosis* DNA repair.

To include time-delays, we used a skip-chain model. The Viterbi shortest path allowed us to choose between time delayed interactions of two genes of same and different time delays. This lets us identify the best interaction information from the dataset. By using a single parent Viterbi path to model the upregulated events, we were able to focus on special cases in the DBN. This significantly reduces the search space for the GA. Our search is however constrained by various parameters like MI and number of parents.

Lastly, using priors gave us higher likelihood and improved the over-fitting in building the regulatory networks. The Dirichlet prior gave fewer hubs and higher likelihood. We can see that the prediction with prior allows higher-orders of the linear model as well.

Chapter 5

Discussion and Conclusions

The different models explained in the previous chapters can be used depending on the availability of data and number of genes. In this chapter, we compare and discuss the utility of the three models : dynamic Bayesian networks, higher-order dynamic Bayesian networks and skip-chain dynamic Bayesian networks. We compare them with respect to the following aspects (1) use of priors in different models namely Dirichlet for DBN, connectivity prior based on mutual information for HDBN and lastly an adaptive conjugate prior for skip-chain model (2) we compare stability of different models including stability in-between delays for HDBN and skip-chain model (3) next we compare application of probabilistic residual bootstrap to assess confidence of networks in different delay models and its ability to handle the limited time points (4) lastly, we discuss issues in validation of different models such as effect of number of time points and number of nodes and availability of protein interaction data.

We demonstrate the comparison by using synthetic as well as real data obtained in yeast cell-cycle, *M.tuberculosis* and human Hela cell-cycle which have been previously predicted. Our experiments show that our models yield regulatory networks which are biologically plausible and the time-delays can give more insight into nature of function. We look into key genes and hubs predicted across different models.

We begin the chapter with a comparison of effect of prior on prediction in different models on synthetic networks. Next, we compare the effect of increasing number of genes and time points. This is followed by stability of networks, stability between similar delays of the same edge and comparison of the probabilistic bootstrap on synthetic data with delays. Lastly, we look at real dataset and biological issues in validating delays.

The above comparisons are followed by a discussion section on biological relevance of various aspects of the thesis including stability, delays and optimization. We conclude the thesis by looking at limitations as well as directions for future work.

5.1 Comparison of different Models

5.1.1 Priors on Models

Introduction of priors can improve the robustness of the parameters and overfitting of the data. A BN assumes that parameters associated with each state of the parents of a gene are independent. We can apply a Dirichlet distribution as a prior to the multinomial distribution of each gene and parent state. It uses pseudo counts to represent missing samples and hence smooth the probability distributions. When the data set is small which is often the case with microarray data, the Dirichlet prior performs well. If the number of time samples is large it might over-penalize the graph. In addition to the parameter prior we also use a topology prior where number of parents of a gene is constrained to follow a binomial distribution where mean number of parents can be pre-specified, this ensures that the MCMC does not get stuck in a local minima due to too many or too few parents for a gene. We show that increasing Dirichlet prior leads to improved prediction after which further increase causes a saturation to occur. High value of mean number of parents can however result in a lower accuracy and unstable convergence.

In the case of VDBN we consider higher-order edges, the number of genes in the network is $(r + 1) \times n$ where r is the order of the VDBN. The MCMC spans increased complexity landscapes, we resort to a reversible jump MCMC to determine optimal order of an edge. To ensure detailed balanced condition in RJMCMC the forward probability of increasing the order must be the same as backward probability of reducing the order of delay. The probability of selecting r ordered nodes follows a uniform distribution. We include a mutual information threshold to help speedup the MCMC. It can be seen that accuracy improves with MI threshold, however very high MI can result in too few edges or poor convergence. It can also be seen that increasing or decreasing order of an edge using RJMCMC significantly helps with faster convergence.

For long delays we think an adaptive prior is more suitable, since the pathway is very sensitive to changes. We use the reliable lower order probability as conjugate prior to the volatile feature specific Viterbi forward probability to

approximate long delays. Hence, the interpolation constant varies depending on interaction and depends on total probability of all possible paths between genes. In summary, the different priors are suitable for different types of delays and inappropriate priors could lead to excess penalty and poor convergence.

5.1.2 Effect of Number of Samples and Genes

We compare the first-order DBN with ten genes and 100 time samples with VAR and GBN when data is generated from the respective models. The VAR model performs well on small networks with single time point delay. The DBN needs more time samples compared to the VAR model. Next, we check the scalability of the DBN to number of genes and available time samples. It can be seen with number of genes equal to number of time samples we can get a reasonable accuracy using the discrete DBN model. Our simulation also indicates that small networks required more number of time points, relative to the number of nodes in the network, than the large networks which tend to converge more easily.

For higher-order model we compared a network of size ten and up-to three delays with VAR and GBN. Here the VAR needed more time samples than VDBN to correctly recover the network. GBN performance was good when delays are included and similar to VDBN. We also studied the scalability of the model to higher delays from two to four as well as larger networks up to 100 genes. FN are higher for longer delays, this suggests that longer delays are more difficult to predict in VDBN compared to smaller delays.

Lastly, we simulated synthetic networks with long delays of order four to seven and fitted to the skip-chain model, VAR and GBN. While VAR did not recover any edges, GBN showed good prediction accuracy which did not improve with number of time samples. For larger networks the skip-chain model outperformed the VDBN even in long delays. This seems to be a significant contribution as by modeling long delays as cascades of short-delays we can predict more accurate and also stable pathways.

5.1.3 Stability of the Networks

Stability of building GRN was evaluated by drawing synthetic time-series gene expressions from a network of known topology. The stability criterion was measured by the mean over all pairwise Hamming distances among all the detected connections over different datasets. The stability of DBN decreases with the number of time points relative to the network size. The higher-order

DBN is more unstable than the first-order DBN for the same number of genes. The stability lowers further as we increase order from two to four, however the decrease is not significant. While increasing number of time samples only slightly improved stability of DBN, the stability of VDBN increases almost linearly with number of time samples. Hence, predicting delays in the network required more time samples than just predicting first-order interactions.

We also analysed how the delay of an edge was varying over different simulation or in other words the stability of the delay of an edge. The small network of ten genes and larger network of 100 genes have similar stability. However, networks of moderate size around 20 genes showed a decline in stability with increasing delay for the fourth-order VDBN, this could be since larger networks are able to stabilize faster than intermediate size networks.

The skip-chain model had higher stability corresponding to the higher accuracy of the networks. The stability also increased more sharply with number of time samples. For the 20 gene network increase in stability in higher delays was very low similar to the VDBN strongly suggesting that the network size effects the prediction algorithm.

5.1.4 Bootstrapping Models

The idea behind bootstrapping is repeated estimation of a statistics from observed data. Because, of the limited number of time samples and in-order to evaluate the confidence of the networks and connections derived, we resort to bootstrapping. Vector sieve bootstrap generates samples from an empirical distribution derived from a distribution of residuals. Since, we discretized data so that we can fit a multinomial fitness function, hence our distribution of errors is also multinomial. It can be easily estimated by calculating probability tables using predicted network. The data is then fitted to this CPD to generated tables for errors in the data. It can be shown that residual bootstrap performs slightly better than corresponding vector sieve bootstrap for DBN. We extended the concept to include errors for delayed interactions. While vector sieve performs very poorly on delays, the probabilistic residual bootstrap continues to give good improvement in prediction for higher-order DBN, as well as skip-chain model.

5.1.5 Validation of Real datasets

We validated our predicted networks by employing protein-protein interaction (PPI) data. Since, information on delays is not easily available hence we

validate higher-delay interactions by looking at cascades of genes in PPIN corresponding to a predicted edge. We call this k -skip validation. Since, genes are involved in multiple-interleaving activities, we look at higher-order PPI edges while validating networks. This is especially useful for large networks. We show that the DBN performs slightly better than the VAR model especially for large network. Using HDBN and skip-chain models respectively lead to significant increase in accuracy. For large network of Hela genes since information on delays is not available we use the k -skip model. It can be seen that the skip-chain model is better suited than DBN when validation is done using k -skip model. We also compared top ten hubs by various models in Hela genes. Hubs could be mapped to nature of activities such as gradual or instantaneous regulation in the pathway.

5.2 Discussion

GRN's are a fundamental feature of all living cells. They have diverse sub-modules. Hence predicting reliable GRN's will require a very flexible model which can fuse data from various sources. The microarray data is able to capture a picture of the instantaneous network. Here genes having similar expression levels across the time series are most probably interacting. However this is insufficient information. Adding other data sources like prior PPIN greatly improves the prediction power. This indicates that data fusion is very critical while predicting a network.

In particular we notice that time-delayed interactions are very characteristic of biological systems. It is important to model them appropriately. This improves the prediction significantly. Another feature is cross-talk. This maintains the robustness of GRN's in response to stress. A model which allows all possible interactions is hence needed.

A major constraint is the size of the network. The sensitivity goes down for large networks. This is because we make an assumption that network size is small. With increase in number of genes entropy of the system goes up and all interactions become equally likely. Next, it can be seen that even for a small genome like yeast, the computation of network can take a few hours. Optimization is hence needed to decrease the computational complexity.

The overall research objective is to develop a new framework which is suitable for accurately predicting biological pathways and identifying underlying modules. It has been shown that the current techniques are not well suited for such tasks, this new framework will be using a novel way of predicting interactions from time-series expression profiles.

5.2.1 Stability of Models

Biological networks are constantly subjected to random perturbations and noise, and efficient feedback and compensatory mechanism naturally provide stability. Gene expression data gathered from microarray are confounded by not only biological noise and variations but also measurement errors and artifacts. As seen, although the accuracy is high, GRN built from time-series of gene expressions is often unstable or non-reproducible.

In this thesis, we investigated stability of building GRN in DBN framework by using simulations. By bootstrapping gene expression time-series datasets synthetically from scale-free networks (taken as topology closer to biological networks), stability of building GRN of different sizes, with varying number of time points were studied. Although a number of time points equal to the number of genes in the network provide accuracy as high as 90%, presence of false negatives and false positives were high. This could only be reduced by increasing the number of time points at which gene expressions are gathered.

This work with synthetic data provides a foundation for future studies of stability of building GRN with real data. In practice, although gene expressions of a large number of genes are gathered simultaneously, repetition over many time points is limited due to high cost and time involved, and due to the nature of experiments. In such scenarios, either bootstrapping or regularization of time samples of gene expressions is needed to study the stability and reproducibility of computationally derived GRN. Investigation on real data is beyond the scope of the present manuscript but the application to real data remains a high priority challenge in functional genomics and systems biology.

5.2.2 Higher-order delays

Knowledge of regulatory networks gives a comprehensive picture of how genes function and interact with one another to execute a specific biological function. Earlier methods used for modeling gene regulatory networks assumed that the regulatory events are synchronous or had a fixed maximum delay.

Pathways are often triggered by transcription factors which in turn express genes and produce proteins. Therefore, the regulatory interactions in molecular pathways can be given by GRN. Gene regulations generally includes dynamic feedback loops, cascaded interactions, intermediary factors, etc., which provides for underlying biological mechanisms of regulation. This results in different time delays in regulatory interactions.

Our experiments suggest that the present framework generates biologically more meaningful regulatory networks. This is expected as the regulatory interactions in biological pathways are inherently asynchronous. Validation of our method was limited because the delays of interactions in cell cycle regulations are unknown by experimental means. However, VDBN technique showed better fit to the data than DBN and validation with PPI data showed more accuracy.

5.2.3 Long delays

We have considered higher-order DBN (HDBN) for representing delays in regulations. When larger delays are involved, implementation of HDBN becomes intractable. Therefore, we proposed a skip-chain HDBN. This involved two components: linear model to represent short delays and skip model to represent long delays. These two components may represent actions of activator and inhibitor involved in regulatory interactions. Our method was evaluated against earlier approaches, which shows our method better fits the gene expression data when GRN was built. In order to provide a more biologically meaningful validation, we performed comparison with the protein-protein interaction data. That validation also showed superiority of our technique over other methods but because of the incompleteness of PPI data sources, such comparisons results in large number of false positives.

Skip-chain models address the difficulties of a HDBN by easily incorporating long time-delayed regulations. The skip model is a first-order HMM and captures long-distance dependencies of input time-course gene expressions. This inference technique leads to lower computational times without loss in the accuracy compared to HDBN. The forward Viterbi path through the trellis determines the best long-distant time delay and therefore automatically finds the best higher-order interactions between genes. The method gives more accurate and biologically plausible networks than DBN implementations as short- and long- time delays are inherent in biological interactions.

5.2.4 Optimization

A small perturbation in gene expressions significantly alters the topology of the network and strengths of regulatory interactions. Therefore, stability is an important characteristic of algorithms building GRN. Our study on sparse DBN models of GRN indicates that GA are computationally faster in the prediction of regulatory interactions but have inherently poor stability compared to MCMC.

In order to account for delays, we introduce a probabilistic framework to generate HDBN which automatically finds the order of the network as well as the optimal structure. While a first-order Bayesian network has a perfectly convex solution space introduction of higher-orders introduces several local minima. Hence the order of the network was found by using a new VDBN framework derived by a variable-order MCMC technique.

5.3 Limitations and Future Work

In this thesis a new implementation of dynamic Bayesian networks with long time-delays has been proposed for gene regulatory networks. On synthetic data we have shown that the proposed model out performs vector autoregressive models in predicting delays and interactions in GRN. However, validation of the study is limited by available information on true interactions and associated time delays.

GRN are often very complex with single genes being regulated by hundreds of other genes. Since, complexity of Bayesian networks increases exponentially with number of parents we had to use a topology prior to restrict the average number of parents in the architecture so as to render a stable structure. Convergence of large networks was achieved using multiple starting structure and bootstrapping the samples. Significant work needs to be done on the large networks in terms of accuracy and computational speeds. GPU based implementations have to be considered to reach global maxima when number of genes are large.

Higher-order dependencies are significant for time-series gene expression data analysis when deriving gene regulatory networks. Future work would be focused on studying the stability between these delays and its implications on the predicted network. Feature functions are a good way to include prior knowledge into a gene-regulatory network and the method is found to be robust when applied to larger networks. The approach is computationally efficient however here, we have only considered interactions where both genes are found over-expressed at long delays.

Similar modeling can be done for features based on function, location and nature of interaction such as activation, inhibition and feedback events for causal networks of genes. The probabilistic residual bootstrap helps determine the distribution of errors. Further, work needs to address how this can be integrated into the fitness function. Key areas of research following this thesis can be classified as those related to structure and implementation while others relating to acquisition of suitable feature sets.

5.3.1 GRN framework and Implementation

Extension to conditional random fields integrating several biological datasets will be crucial to ensure global convergence of Bayesian networks. Comparison with recent methods with multiple transition matrices for different modes can be done. Quasi-newton methods such as Broyden-Fletcher-Goldfarb-Shanno (BFGS) are well-known algorithms for approximating log-linear functions where the objective function is convex. They are based on Newton's method to find a stationary point of a function, where the gradient is zero and require only a few iterations to converge. A comparison or hybrid model with the VM-CMC will be a part of our future work. The prediction is poor as network size increases; a possible solution might be to break down the big network and merge sub-structures again after prediction.

5.3.1.1 Missing value estimation

The software KNN-Impute allows to choose among different distance measures. Several authors have evaluated the software previously. Following Stanford Microarray Database (SMD) we have used Euclidean distance on our raw microarray datasets. Benchmark datasets have only few missing values since replicates are available. Future work can certainly explore different distance measures and effect on number of missing values.

5.3.2 Acquisition of Feature Sets

Currently, we are considering synthetic as well as benchmark datasets from three previous authors focusing on cell-cycle and DNA replication. It would be interesting to investigate other pathways and diseases found at NCBI Gene expression omnibus and provide a consolidated database to compare and visualize the predicted networks. It is important to identify correct feature sets for gene networks to generate reliable networks. In particular we need to look at activation and inhibition events.

Here we have only considered two types of data, microarray and PPI to check the validity of our prediction. Adding other types of data like functional annotation given by COG (Clusters of Orthologous proteins) will have to be explored. This will help us develop a framework for reliable pathways where fusion of different types of data is possible. A very common event in gene networks is complex formation. We would like to include it in our model in future. Also, we need to test separately on experimental and literature derived interactions. Relating phenotypes to disease diagnosis remains a big challenge.

Clinicians are now resorting to genotype to validate symptoms by looking at abnormal expression of certain genes in patients.

5.3.2.1 Biologically relevant Priors

Among other approaches, Bayesian networks have been the most mature frameworks for integration of heterogeneous data in the form of prior. A commonly used prior for including biological knowledge is the Gibbs prior [54]. Here additional information is provided as a prior probability of the network. This strategy has shown an increase in accuracy of GRN reconstruction. Prior knowledge can include protein-protein interaction, transcription-factor DNA binding, sequence motifs, pathways, literature mining etc. Hence, future directions will focus on experiments with different biologically inspired priors for the dynamic networks.

Dirichlet prior used in our model are still rigid as a uniform expected sample size is assumed. It would be interesting to experiment the effect of adaptive priors which are flexible and hence biologically relevant.

5.3.2.2 Time-delay Information

Time-delay information is certainly available on some small benchmark networks such as yeast cell-cycle. However, biological databases such as BioGrid currently only store information on interactions and do not give us information of the delay which has to be inferred from literature, hence we proposed a k -skip validation for edges with unknown delays. Recently several authors are focusing on extracting action and time information through text-mining of publications cited in Biogrid. Future work will include integration of such information into the prediction and validation algorithm.

References

- [1] Breitkreutz BJ, Stark C, Reguly T, Boucher L, Breitkreutz A, Livstone M, Oughtred R, Lackner DH, Bahler J, Wood V, Dolinski K, Tyers M: **The BioGRID Interaction Database: 2008 update.** *Nucl. Acids Res.* 2008, **36**(suppl1):D637–640.
- [2] Chen Y, Han K: *Dynamic Identification and Visualization of Gene Regulatory Networks from Time-Series Gene Expression Profiles Emerging Intelligent Computing Technology and Applications*, Springer Berlin / Heidelberg, *Volume 5754 of Lecture Notes in Computer Science* 2009 :65–74.
- [3] Pauplin O, Jiang J: *A Dynamic Bayesian Network Based Structural Learning towards Automated Handwritten Digit Recognition Hybrid Artificial Intelligence Systems*, *Volume 6076 of Lecture Notes in Computer Science* 2010 :120–127.
- [4] Nagele A, Dejori M, Stetter M: *Robust Learning of High-dimensional Biological Networks with Bayesian Networks Robust Intelligent Systems*, Springer London 2008 :139–161.
- [5] Wu CC, Huang HC, Juan HF, Chen ST: **GeneNetwork: an interactive tool for reconstruction of genetic networks using microarray data.** *Bioinformatics* 2004, **20**(18):3691–3.
- [6] Mooij JM: **libDAI: A Free and Open Source C++ Library for Discrete Approximate Inference in Graphical Models.** *Journal of Machine Learning Research* 2010, **11**(8):2169–2173.
- [7] Niculescu RS, Mitchell TM, Rao RB: **Bayesian Network Learning with Parameter Constraints.** *J. Mach. Learn. Res.* 2006, **7**:1357–1383.

- [8] Galan S, Arroyo-Figueroa G, Diez F, Sucar L: *Comparative Evaluation of Temporal Nodes Bayesian Networks and Networks of Probabilistic Events in Discrete Time MICA I 2004: Advances in Artificial Intelligence*, Springer Berlin / Heidelberg, *Volume 2972 of Lecture Notes in Computer Science 2004* :498–507.
- [9] Nir Friedman MG, Abraham W: **Data Analysis with Bayesian Networks: A Bootstrap Approach** 1999.
- [10] Zhengzheng X, Dan W: **Modeling Multiple Time Units Delayed Gene Regulatory Network Using Dynamic Bayesian Network**. In *Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on* 2006:190–195.
- [11] Ling CX, Huajie Zhang CX, Brodley CE, Danyluk A: **The Representational Power of Discrete Bayesian Networks**. *Journal of Machine Learning Research* 2003, **3**(4/5):709–721.
- [12] Green PJ: **Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination**. *Biometrika* 1995, **82**(4):711–732.
- [13] Galley M: **A Skip-Chain Conditional Random Field for Ranking Meeting Utterances by Importance**. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)* 2006:364–372.
- [14] Feng L, Kun T, Khan L: **Hierarchical clustering of gene expression data**. In *Bioinformatics and Bioengineering, 2003. Proceedings. Third IEEE Symposium on* 2003:328–335.
- [15] Wu W, Liu X, Xu M, Peng J, Setiono R: **A hybrid SOM-SVM method for analyzing zebra fish gene expression**. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, Volume 2* 2004:323–326 Vol.2.
- [16] Alter O, Brown PO, Botstein D: **Singular value decomposition for genome-wide expression data processing and modeling**. *Proc Natl Acad Sci U S A* 2000, **97**(18):10101–6.
- [17] Lee SI, Batzoglou S: **Application of independent component analysis to microarrays**. *Genome Biol* 2003, **4**(11):R76.

- [18] Liang S, Fuhrman S, Somogyi R: **Reveal, a general reverse engineering algorithm for inference of genetic network architectures.** *Pac Symp Biocomput* 1998, :18–29.
- [19] Li P, Zhang C, Perkins EJ, Gong P, Deng Y: **Comparison of probabilistic Boolean network and dynamic Bayesian network approaches for inferring gene regulatory networks.** *BMC Bioinformatics* 2007, **8**:S13–S20.
- [20] Nariai N, Kim S, Imoto S, Miyano S: **Using protein-protein interactions for refining gene networks estimated from microarray data by Bayesian networks.** *Pac Symp Biocomput* 2004, :336–47.
- [21] Imoto S, Kim S, Goto T, Miyano S, Aburatani S, Tashiro K, Kuhara S: **Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network.** *J Bioinform Comput Biol* 2003, **1**(2):231–52.
- [22] Liu B, Thiagarajan P, Hsu D: **Probabilistic Approximations of Signaling Pathway Dynamics.** In *Computational Methods in Systems Biology* 2009:251–265.
- [23] Gebert J, Motameny S, Faigle U, Forst CV, Schrader R: **Identifying Genes of Gene Regulatory Networks Using Formal Concept Analysis.** *Journal of Computational Biology* 2008, **15**(2):185–194.
- [24] Friedman N, Linial M, Nachman I, Pe’er D: **Using Bayesian Networks to Analyze Expression Data.** *Journal of Computational Biology* 2000, **7**(3-4):601–620.
- [25] Segal E, Wang H, Koller D: **Discovering molecular pathways from protein interaction and gene expression data.** *Bioinformatics* 2003, **19 Suppl 1**:i264–71.
- [26] Pei J, Jiang D, Zhang A: **Mining cross-graph quasi-cliques in gene expression and protein interaction data.** In *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on* 2005:353–356.
- [27] Tu K, Yu H, Li YX: **Combining gene expression profiles and protein-protein interaction data to infer gene functions.** *J Biotechnol* 2006, **124**(3):475–85.

- [28] Yates F: **The analysis of replicated experiments when the field results are incomplete.** *Empire J. Experimental Agriculture* 1933, **1**:129–142.
- [29] Wilkinson GN: **Estimation of Missing Values for the Analysis of Incomplete Data.** *Biometrics* 1958, **14**(2):257–286.
- [30] Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, Botstein D, Altman RB: **Missing value estimation methods for DNA microarrays.** *Bioinformatics* 2001, **17**(6):520–525.
- [31] Sherlock G, Hernandez-Boussard T, Kasarskis A, Binkley G, Matese JC, Dwight SS, Kaloper M, Weng S, Jin H, Ball CA, Eisen MB, Spellman PT, Brown PO, Botstein D, Cherry JM: **The Stanford Microarray Database.** *Nucleic Acids Research* 2001, **29**:152–155.
- [32] Cappe O, Moulines E, Ryden T: *Inference in hidden Markov models* 2005.
- [33] Christopher JJ, Christian WH: **Independent component analysis for biomedical signals.** *Physiological Measurement* 2005, **26**:R15.
- [34] Breitkreutz BJ, Stark C, Reguly T, Boucher L, Breitkreutz A, Livstone M, Oughtred R, Lackner DH, Bahler J, Wood V, Dolinski K, Tyers M: **The BioGRID Interaction Database: 2008 update.** *Nucl. Acids Res.* 2008, **36**(suppl1):D637–640.
- [35] Friedman N, Murphy K, Russell S: **Learning the Structure of Dynamic Probabilistic Networks.** *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)* 1998, :139–14.
- [36] Chaturvedi I, Rajapakse J: **Fusion of Gene Regulatory and Protein Interaction Networks Using Skip-Chain Models.** In *Pattern Recognition in Bioinformatics, Volume 5265 of Lecture Notes in Computer Science* 2008:214–224.
- [37] Akutsu T, Miyano S, Kuhara S: **Algorithms for Identifying Boolean Networks and Related Biological Networks Based on Matrix Multiplication and Fingerprint Function.** *Journal of Computational Biology* 2000, **7**(3-4):331–343.

- [38] Ota K, Yamada T, Yamanishi Y, Goto S, Kanehisa M: **Comprehensive Analysis of Delay in Transcriptional Regulation Using Expression Profiles** 2003.
- [39] Husmeier D: **Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks.** *Bioinformatics* 2003, **19**(17):2271–82.
- [40] Tie-Fei L, Wing-Kin S, Ankush M: **Learning Multi-Time Delay Gene Network Using Bayesian Network Framework** 2004. [1033661 640-645].
- [41] Bastos G, Guimaraes KS: **A simpler Bayesian network model for genetic regulatory network inference.** In *Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on, Volume 1* 2005:304–309 vol. 1.
- [42] Buhlmann P: **Bootstraps for Time Series.** *Statistical Science* 2002, **17**:52–72.
- [43] Schulz T, Radliski L, Gorges T, Rosenstiel W: **Predicting the Flow of Defect Correction Effort using a Bayesian Network Model.** *Empirical Software Engineering*:1–43.
- [44] Aho HJ, Ullman J, Aho J: *Data Structures and Algorithms*. AddisonWesley: Reading, MA. 1983.
- [45] Ram R, Chetty M: **A guided genetic algorithm for learning gene regulatory networks.** In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on* 2007:3862–3869.
- [46] Corne D, Pridgeon C: **Investigating issues in the reconstructability of genetic regulatory networks.** In *Evolutionary Computation, 2004. CEC2004. Congress on, Volume 1* 2004:582–589 Vol.1.
- [47] Bender C: **Systematic analysis of time resolved high-throughput data using stochastic network inference methods.** *PhD thesis*, Ruperto-Carola University 2011.
- [48] Yardim C, Gerstoft P, Hodgkiss WS, Huang CF: **Refractivity from clutter (RFC) estimation using a hybrid genetic algorithm-Markov chain Monte Carlo method.** In *Antennas and Propagation Society International Symposium, 2005 IEEE, Volume 1B* 2005:743–746 vol. 1B.

- [49] Wagner J, Stolovitzky G: **Stability and Time-Delay Modeling of Negative Feedback Loops.** *Proceedings of the IEEE* 2008, **96**(8):1398–1410.
- [50] Heckerman D, Geiger D, Chickering DM: **Learning Bayesian Networks: The Combination of Knowledge and Statistical Data.** *Machine Learning* 1995, **20**(3):197–243.
- [51] Steck H, Jaakkola TS: **On the Dirichlet Prior and Bayesian Regularization** 2002, :697–704.
- [52] Werhli AV, Husmeier D: **Reconstructing Gene Regulatory Networks with Bayesian Networks by Combining Expression Data with Multiple Sources of Prior Knowledge.** *Statistical Applications in Genetics and Molecular Biology* 2007, **6**.
- [53] Huang Y, Wang J, Zhang J, Sanchez M, Wang Y: **Bayesian Inference of Genetic Regulatory Networks from Time Series Microarray Data Using Dynamic Bayesian Networks.** *Journal of Multimedia* 2007, **2**:46–56.
- [54] Husmeier D, Werhli AV: **Bayesian integration of biological prior knowledge into the reconstruction of gene regulatory networks with Bayesian networks.** *Computational systems bioinformatics* 2007, **6**:85–95.
- [55] Chaturvedi I, Sakharkar M, Rajapakse J: **Validation of Gene Regulatory Networks from Protein-Protein Interaction Data: Application to Cell-Cycle Regulation.** In *Pattern Recognition in Bioinformatics, Volume 4774 of Lecture Notes in Computer Science* 2007:300–310.
- [56] Shen-Orr SS, Milo R, Mangan S, Alon U: **Network motifs in the transcriptional regulation network of Escherichia coli.** *Nat Genet* 2002, **31**:64–68. [10.1038/ng881].
- [57] Xiaohua H, Michael N, Fang-Xiang W, Bahrad AS: **Mining, modeling, and evaluation of subnetworks from large biomolecular networks and its comparison study.** *Trans. Info. Tech. Biomed.* 2009, **13**(2):184–194. [1651249].

- [58] Prinzie A, Van den Poel D: *Dynamic Bayesian Networks for Acquisition Pattern Analysis: A Financial-Services Cross-Sell Application New Frontiers in Applied Data Mining*, Springer Berlin / Heidelberg, Volume 5433 of Lecture Notes in Computer Science 2009 :123–133.
- [59] Wagner J, Ma L, Rice JJ, Hu W, Levine AJ, Stolovitzky GA: **p53-Mdm2 loop controlled by a balance of its feedback strength and effective dampening using ATM and delayed feedback.** *Systems Biology, IEE Proceedings* 2005, **152**(3):109–118.
- [60] Friedman N, Murphy K, Russell S: **Learning the Structure of Dynamic Probabilistic Networks.** *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)* 1998, :139–14.
- [61] Perrin BE, Ralaivola L, Mazurie A, Bottani S, Mallet J, dAlchBuc F: **Gene networks inference using dynamic Bayesian networks.** *Bioinformatics* 2003, **19**(suppl 2):ii138–ii148.
- [62] Castillo E, Gutierrez JM, Hadi AS: *Expert systems and probabilistic network models* 1996.
- [63] Qiu P, Gentles AJ, Plevritis SK: **Fast calculation of pairwise mutual information for gene regulatory network reconstruction.** *Computer methods and programs in biomedicine* 2009, **94**(2):177–180.
- [64] Zhengzheng X, Dan W: **Modeling Multiple Time Units Delayed Gene Regulatory Network Using Dynamic Bayesian Network.** In *Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on* 2006:190–195.
- [65] Qiu P, Gentles AJ, Plevritis SK: **Reducing the Computational Complexity of Information Theoretic Approaches for Reconstructing Gene Regulatory Networks.** *Journal of Computational Biology* 2010, **17**(2):169–176.
- [66] Rencher AC: *Methods of multivariate analysis*. 2nd edition 2002.
- [67] Harrison L, Penny WD, Friston K: **Multivariate autoregressive modeling of fMRI time series.** *NeuroImage* 2003, **19**(4):1477–1491. [Doi: DOI: 10.1016/S1053-8119(03)00160-5].
- [68] Castillo E, Kjrulff U: **Sensitivity analysis in Gaussian Bayesian networks using a symbolic-numerical technique** 2003.

- [69] Van Der Laan MJ, Bryan J: **Gene expression analysis with the parametric bootstrap.** *Biostatistics* 2001, **2**(4):445–461.
- [70] Buhlmann P: **Blockwise Bootstrapped Empirical Process for Stationary Sequences.** *The Annals of Statistics* 1994, **22**(2):995–1012.
- [71] Ruiz M, Girn FJ, Prez CJ, Martn J, Rojano C: **A Bayesian model for multinomial sampling with misclassified data.** *Journal of Applied Statistics* 2008, **35**(4):369–382.
- [72] Huang B: **Multivariable model validation in the presence of time-variant disturbance dynamics.** *Chemical Engineering Science* 2000, **55**(20):4583–4595.
- [73] Chushin K, Fang-Xiang W, Gopalan S, Anthony JK: **Using a state-space model and location analysis to infer time-delayed regulatory networks.** *EURASIP J. Bioinformatics Syst. Biol.* 2009, **2009**:1–1. [1734749].
- [74] Albert-Laszlo B, Reka A: **Emergence of scaling in random networks.** *Science* 1999, **286**(5439):509.
- [75] Fujita A, Sato J, Garay-Malpartida H, Yamaguchi R, Miyano S, Sogayar M, Ferreira C: **Modeling gene expression regulatory networks with the sparse vector autoregressive model.** *BMC Systems Biology* 2007, **1**:39.
- [76] Gotshall S, Rylander B: **Optimal Population Size and the Genetic Algorithm.**
- [77] Oliver SG: **Functional genomics: lessons from yeast.** *Philos Trans R Soc Lond B Biol Sci* 2002, **357**(1417):17–23.
- [78] Cho R, Campbell M, Winzeler E, Steinmetz L, Conway A, Wodicka L, Wolfsberg Ta: **A genome-wide transcriptional analysis of the mitotic cell cycle.** *Molecular Cell* 1998, **2**:65–73.
- [79] Boshoff HIM, Myers TG, Copp BR, McNeil MR, Wilson MA, Barry CE: **The Transcriptional Responses of Mycobacterium tuberculosis to Inhibitors of Metabolism.** *Journal of Biological Chemistry* 2004, **279**(38):40174–40184.

- [80] Dullaghan EM, Brooks PC, Davis EO: **The role of multiple SOS boxes upstream of the Mycobacterium tuberculosis *lexA* gene identification of a novel DNA-damage-inducible gene.** *Microbiology* 2002, **148**(11):3609–3615.
- [81] Radde N, Gebert J, Forst CV: **Systematic component selection for gene-network refinement.** *Bioinformatics* 2006, **22**(21):2674–2680.
- [82] Kazemy A, Hosseini SA, Farrokhi M: **Second Order Diagonal Recurrent Neural Network.** In *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on*:251–256.
- [83] Shi J, Murray-smith R: **Birth-death MCMC methods for mixtures with unknown number of components** 2002.
- [84] Jiayin W, Yufei H, Sanchez M, Yufeng W, Jianqiu Z: **Reverse Engineering Yeast Gene Regulatory Networks using Graphical Models.** In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on, Volume 2* 2006:II–II.
- [85] Bhatnagar S, Borkar VS, Akarapu M, Mannor S: **A Simulation-Based Algorithm for Ergodic Control of Markov Chains Conditioned on Rare Events.** *Journal of Machine Learning Research* 2006, **7**(10):1937–1962.
- [86] Robinson JW, Hartemink AJ, Ghahramani Z: **Learning Non-Stationary Dynamic Bayesian Networks.** *Journal of Machine Learning Research* 2010, **11**(12):3647–3680.
- [87] Roucos S, Makhoul J, Schwartz R: **A variable-order Markov chain for coding of speech spectra.** In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '82., Volume 7*:582–585.
- [88] Seth AK: **A MATLAB toolbox for Granger causal connectivity analysis.** *Journal of Neuroscience Methods* 2010, **186**(2):262–273.
- [89] Baccal LA, Sameshima K: **Partial directed coherence: a new concept in neural structure determination.** *Biological Cybernetics* 2001, **84**(6):463.

- [90] Kelly OE, Johnson DH: **Adaptive channel equalization using context trees**. In *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on, Volume 3*:1949–1952 vol.3.
- [91] Iti Chaturvedi JCR: **Gene regulatory networks with variable order dynamic Bayesian networks**. *WCCI 2010* 2010.
- [92] Lukic AS, Wernick MN, Galatsanos NP, Yongyi Y: **Reversible jump Markov chain Monte Carlo for brain activation detection**. In *Statistical Signal Processing, 2003 IEEE Workshop on*:506–509.
- [93] Denison DGT, Mallick BK, Smith AFM: **Automatic Bayesian curve fitting**. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 1998, **60**(2):333.
- [94] Orosz G, Moehlis J, Murray RM: **Controlling biological networks by time-delayed signals**. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 2010, **368**(1911):439–454.
- [95] Jun Z, Zaiqing N, Bo Z, Ji-Rong W: **Dynamic Hierarchical Markov Random Fields for Integrated Web Data Extraction**. *Journal of Machine Learning Research* 2008, **9**(7):1583–1614.
- [96] Tsungnan L, Horne BG, Tino P, Giles CL: **Learning long-term dependencies in NARX recurrent neural networks**. *Neural Networks, IEEE Transactions on* 1996, **7**(6):1329–1338.
- [97] Shuanhu B, Haizhou L: **Bayesian Learning of N-Gram Statistical Language Modeling**. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on, Volume 1* 2006:I–I.
- [98] Meyer-Baese A, Koshkouei AJ, Emmett MR, Goodall DP: **Global stability analysis and robust design of multi-time-scale biological networks under parametric uncertainties**. *Neural Networks* 2009, **22**(56):658–663.
- [99] Quanxin Z, Jinde C: **Robust Exponential Stability of Markovian Jump Impulsive Stochastic Cohen-Grossberg Neural Networks With Mixed Time Delays**. *Neural Networks, IEEE Transactions on* 2010, **21**(8):1314–1325.

- [100] Lebre S, Becq J, Devaux F, Stumpf M, Lelandais G: **Statistical inference of the time-varying structure of gene-regulation networks.** *BMC Systems Biology* 2010, **4**:130.
- [101] Grzegorzczak M, Husmeier D: **Improvements in the reconstruction of time-varying gene regulatory networks: dynamic programming and regularization by information sharing among genes.** *Bioinformatics* 2011, **27**(5):693–699.
- [102] Kolter JZ, Maloof MA: **Dynamic Weighted Majority: An Ensemble Method for Drifting Concepts.** *Journal of Machine Learning Research* 2007, **8**(12):2755–2790.
- [103] Balasubramaniam P, Rakkiyappan R, Krishnasamy R: **Stochastic stability of Markovian jumping uncertain stochastic genetic regulatory networks with interval time-varying delays.** *Mathematical Biosciences* 2010, **226**(2):97–108.
- [104] Fink GA, service) SO: **Markov Models for Pattern Recognition From Theory to Applications** 2008.
- [105] Hao T, Huang TS: **Improved Graphical Model for Audiovisual Object Tracking.** In *Multimedia and Expo, 2006 IEEE International Conference on* 2006:997–1000.
- [106] Chaturvedi I, Rajapakse J: **Detecting robust time-delayed regulation in Mycobacterium tuberculosis.** *BMC Genomics* 2009, **10**(Suppl 3):S28.
- [107] Kauffman S: **Homeostasis and differentiation in random genetic control networks.** *Nature* 1969, **224**(5215):177–8.
- [108] Shannon CE: *The mathematical theory of communication.* Urbana: : Univ. of Illinois Press 1964. [, by Claude E. Shannon and Warren Weaver.].
- [109] D’Haeseleer P, Liang S, Somogyi R: **Genetic network inference: from co-expression clustering to reverse engineering.** *Bioinformatics* 2000, **16**(8):707–26.
- [110] Tian T, Burrage K: **Stochastic neural network models for gene regulatory networks.** In *IEEE Congress on Evolutionary Computation* 2003:162–169.

- [111] Savageau MA: *Biochemical Systems Analysis: A Study of Function and Design in Molecular Biology*. Addison-Wesley, Reading, Mass. 1976.
- [112] Wang H, Qian L, Dougherty E: **Inference of gene regulatory networks using S-system: a unified approach**. *Systems Biology, IET* 2010, **4**(2):145–156.
- [113] Kojima K: **Estimation of nonlinear gene regulatory networks via L1 regularized NVAR from time series gene expression data**. *Genome Informatics* 2008, **20**:37–51.
- [114] Pearl J: *Probabilistic reasoning in intelligent systems : networks of plausible inference*. The Morgan Kaufmann series in representation and reasoning., San Mateo, Calif.: : Morgan Kaufmann Publishers 1988. [/ Judea Pearl. : ill. ; 24 cm. Includes indexes.].
- [115] Mewes HW, Frishman D, Guldener U, Mannhaupt G, Mayer K, Mokrejs M, Morgenstern B, Munsterkotter M, Rudd S, Weil B: **MIPS: a database for genomes and protein sequences**. *Nucleic Acids Res* 2002, **30**:31–4.

Appendix A

Appendix

A.1 Noise Removal

The linear assumption used in JADE:

$$\widetilde{x(t)} = Ax(t) + \epsilon(t) \quad (\text{A.1})$$

where $x(t)$, $\widetilde{x(t)}$, $\epsilon(t)$ are the true underlying data, the observed data and white noise.

The algorithm first computes whitening matrix W using the sample covariance Σ_0 using the equation $\widetilde{x(t)} = w \times x(t)$. Next, we estimate r covariance matrices Σ_t for different delays $t = \{t'\}_{t'=1}^r$. A joint diagonalization of the matrices is done in $U = \{u_i\}_{i=1}^n$. We can then estimate the mixing matrix $A = W^{-1}U$. Lastly, we recover the true source signals as

$$x(t) = A^T \Sigma_0^{-1} \widetilde{x(t)} \quad (\text{A.2})$$

A.2 Maximum Likelihood estimates for BN

Suppose that the gene expressions are discretized into a set Γ of d levels: $\Gamma = \{k\}_{k=1}^d$. A level of gene expression indicates a state of the gene. Let the set of parent genes regulating the gene i be denoted as a_i and the number of states that the set a_i can take to be q_i . Given the set of conditional distributions with parameters $\theta = \{\theta_i\}_{i=1}^n$, the likelihood can be written as

$$p(x|S) = \int p(x|S, \theta) p(\theta|S) d\theta \quad (\text{A.3})$$

In order to find the likelihood in (A.3) to find the optimal structure, the data is assumed to be either Gaussian or multinomial [62]. Estimation of Gaussian linear structural equations poses serious identifiability problems, hence the

data is usually discretized and a multinomial distribution is used. This is an approximate approach since discretization introduces some error.

Throughout this thesis we assume that the data multinomial because this enables handling of complex interactions of regulatory networks [50]. Let $\theta_{ijk} = p(x_i = k | a_i = j)$ and N_{ijk} be the number of instances of θ_{ijk} that occur in the training data. Using the property of decomposability [60],

$$p(x) = \prod_{i=1}^n \prod_{j=1}^{q_i} \prod_{k=1}^d \theta_{ijk}^{N_{ijk}} \quad (\text{A.4})$$

Such a sequence is a particular type of random sample known as $(d-1)$ -dimensional multinomial sample with parameters $\{\theta_{ijk}\}_{i=1, j=1, k=1}^{n, q_i, d}$. Note that only the counts $N_{ijk} = \sum_{t=1}^T \delta(x_i = k | a_i = j)$ are necessary to determine the posterior from the prior. These counts are said to be "sufficient statistics" for the multinomial sample.

Taking the logarithm of (A.4), the log-likelihood is given by:

$$\log p(x|S, \theta) = \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^d N_{ijk} \log \theta_{ijk} \quad (\text{A.5})$$

where the parameters are constrained by :

$$\sum_{k=1}^d \theta_{ijk} = 1 \forall i, j \in a_i \quad (\text{A.6})$$

By (A.5) with constraints in a Lagrange equation; (A.5):

$$J = \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^d N_{ijk} \log \theta_{ijk} + \sum_{i=1}^n \sum_{j=1}^{q_i} \lambda_{ij} \left(\sum_{k=1}^d \theta_{ijk} - 1 \right) \quad (\text{A.7})$$

Taking the derivative with respect to θ_{ijk} , it can be shown that

$$\frac{\partial J}{\partial \theta_{ijk}} = -\frac{N_{ijk}}{\theta_{ijk}} + \lambda_{ij} = 0 \quad (\text{A.8})$$

$$\frac{N_{ijk}}{\lambda_{ij}} = \theta_{ijk} \quad (\text{A.9})$$

Using (2.9) :

$$\sum_{k=1}^d \theta_{ijk} = 1 \longrightarrow \frac{\sum_{k=1}^d N_{ijk}}{\lambda_{ij}} = 1 \quad (\text{A.10})$$

$$\lambda_{ij} = \sum_{k=1}^d N_{ijk} \quad (\text{A.11})$$

By substituting in (A.9) we get the maximum likelihood estimate of parameters :

$$\hat{\theta}_{ijk} = \frac{N_{ijk}}{\sum_{k=1}^d N_{ijk}} \quad (\text{A.12})$$

By substituting $\hat{\theta}_{ijk}$ in (A.5) , the log-likelihood of the data is given by :

$$\log p(x|S) = \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^d N_{ijk} \log \frac{N_{ijk}}{\sum_{k=1}^d N_{ijk}} \quad (\text{A.13})$$

Maximum likelihood estimates of parameters are known to be good when large amounts of data samples are available. That is, when time-series are long or number of counts are large. However, if the dataset is small, the estimates are sensitive to noise and unreliable due to over fitting.

A.3 Priors of Parameters

By introducing priors to parameter estimation procedure, the robustness of parameters noise and over fitting to data can be improved. The parameters of individual conditional probabilities are assumed independent, which implies global parameter independence. Further, local parameter independence of states is presumed that parameters associated with each state of the parents of a variable are independent. By parameter independence,

$$p(\theta|S) = \prod_{i=1}^n p(\theta_i|S) = \prod_{i=1}^n \prod_{j=1}^{q_i} p(\theta_{ij}|S) = \prod_{i=1}^n \prod_{j=1}^{q_i} \prod_{k=1}^d p(\theta_{ijk}) \quad (\text{A.14})$$

Substituting (2.6) and (2.12) in (2.5) gives :

$$p(x|S) = \prod_{i=1}^n \prod_{j=1}^{q_i} \int \prod_{k=1}^d \theta_{ijk}^{N_{ijk}} p(\theta_{ijk}) d\theta_{ijk} \quad (\text{A.15})$$

Let us assume that the prior distribution of θ_{ijk} is given by Dirichlet dis-

tribution with hyper parameters α_{ijk} :

$$p(\theta_{ijk}|\alpha_{ijk}) = \frac{\Gamma(\sum_k \alpha_{ijk})}{\prod_k \Gamma(\alpha_{ijk})} \prod_{k=1}^d \theta_{ijk}^{\alpha_{ijk}-1} \quad (\text{A.16})$$

From Bayes law :

$$p(\theta|S, x) = \frac{p(x|S, \theta)p(\theta|S)}{\int p(x|S, \theta)p(\theta|S)d\theta} \quad (\text{A.17})$$

where

$$p(\theta|S, x) = \frac{\Gamma(\sum_k \alpha_{ijk} + N_{ijk})}{\prod_k \Gamma(\alpha_{ijk} + N_{ijk})} \prod_{i=1}^n \prod_{j=1}^{q_i} \prod_{k=1}^d \theta_{ijk}^{\alpha_{ijk} + N_{ijk} - 1} \quad (\text{A.18})$$

By substituting (A.16) in (A.17), the likelihood of data is given by : the Bayesian Dirichlet equivalent metric (BDe) can be computed :

$$p(x|S) = \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^d \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \quad (\text{A.19})$$

where $N_{ij} = \sum_{k=1}^d N_{ijk}$, Γ is the gamma function. The above likelihood (A.18) is also known as Bayes Dirichlet metric (BDe) .

Hence unlike BDe, likelihood approximation of the score needs no prior as the posterior probability captures information of the prior. Both models have inherent penalty to avoid over-fitting of data.

For Dirichlet priors the Lagrange function now becomes :

$$J = \sum_{i=1}^n \sum_{j=1}^{q_i} \left(\sum_{k=1}^d N_{ijk} \log \theta_{ijk} + \lambda_{ij} \left(\sum_{k=1}^d \theta_{ijk} - 1 \right) + \sum_{k=1}^d (\alpha_{ijk} - 1) \log \theta_{ijk} \right) \quad (\text{A.20})$$

Taking the derivative with respect to θ_{ijk} :

$$\frac{\partial J}{\partial \theta_{ijk}} = -\frac{N_{ijk}}{\theta_{ijk}} + \lambda_{ij} - \frac{\alpha_{ijk} - 1}{\theta_{ijk}} = 0 \quad (\text{A.21})$$

$$\frac{N_{ijk} + \alpha_{ijk} - 1}{\lambda_{ij}} = \theta_{ijk} \quad (\text{A.22})$$

Using (A.20) :

$$\sum_{k=1}^d \theta_{ijk} = 1 \longrightarrow \frac{\sum_{k=1}^d N_{ijk} + \sum_{k=1}^d \alpha_{ijk} - d}{\lambda_{ij}} = 1 \quad (\text{A.23})$$

$$\lambda_{ij} = \sum_{k=1}^d N_{ijk} + \sum_{k=1}^d \alpha_{ijk} - d \quad (\text{A.24})$$

By substituting in (A.22) we get the maximum likelihood estimate of parameters :

$$\hat{\theta}_{ijk} = \frac{N_{ijk} + \alpha_{ijk} - 1}{\sum_{k=1}^d N_{ijk} + \sum_{k=1}^d \alpha_{ijk} - d} \quad (\text{A.25})$$

A.4 Boolean Networks

The earliest and simplest model of gene regulation was the Boolean network model, introduced by Kauffman [107] in 1969. In Boolean networks gene expression can be quantized into only two states 0 for up-regulation and 1 for down-regulation. The expression state of a gene is functionally related via a logical rule to the expression state of some other genes. For each gene we setup a threshold value and formulate input and output state transitions. The model is defined by a graph s with connectivity matrix $c = \{c_{ii'}\}_{i=1, i'=n}^{i=n, i'=n}$ where $c_{ii'} = 1$ or 0 represents the presents or absence of a connection between the genes and boolean functions that determine the state of a gene i given states of parent gene set a_i .

In a boolean network with maximum in-degree k , the total number of possible rules are 2^{2^k} . Many of these rules do not depend on the inputs. However, rules are detected depending on the inputs and the best way to infer rules with optimal degree k_{eff} is that the optimal structure cannot be reduced to a rule with smaller number of inputs. The most general implementation uses mutual information (MI) [108], [18] as the boolean function to determine minimum k_{eff} and corresponding topology of the graph.

The MI of a gene i with parent set a_i is denoted as $m(i, a_i)$. and is related to the entropy of the gene $h(i)$ and joint entropy of gene and parent set $h(i, a_i)$:

$$m(i, a_i) = h(i) + h(a_i) - h(i, a_i) \quad (\text{A.26})$$

hence if $m(i, a_i) = h(i)$ then $h(a_i) = h(i, a_i)$, where the entropies are calculated from the quantized data :

$$\begin{aligned} h(i) &= - \sum_t \sum_{x_i(t) \in \Gamma} p(x_i(t)) \log p(x_i(t)) \\ h(i, j) &= - \sum_t \sum_{(x_i(t), x_j(t)) \in \Gamma^2} p(x_i(t), x_j(t)) \log p(x_i(t), x_j(t)) \end{aligned} \quad (\text{A.27})$$

Hence $h(a_i) = h(j_1, j_2, \dots, j_{|a_i|})$ where j are parents of gene i in parent set a_i . Similarly, $h(i, a_i) = h(i, j_1, j_2, \dots, j_{|a_i|})$ can be calculated using A.28.

For each gene, more than one regulation combination may confirm MI-analysis. A genetic algorithm is used to search for the optimal structure with a fitness function that minimizes the effective regulation linkage

$$fit = \sum_{i=1}^n (k_{i1} + k_{i2}) \quad (\text{A.28})$$

$k_{i1} = 0$ if $h(a_i) \neq h(i, a_i)$ and a non-zero constant otherwise, $k_{i2} = n - |a_i|$ if $h(a_i) = h(i, a_i)$ and 0 otherwise, here $|a_i|$ is regulation linkage or number of parents of gene i . Here, N individuals which are matrices of the type c are randomly created. One-point mutation and crossover are used to evolve the population.

The Boolean network is mathematically tractable and hence allows investigation of large systems of genes. However, they are not capable of inferring networks with feedback loops and lack causality. The assumption of a single transition rule for each gene becomes problematic with data is noisy and the number of samples is small. Another bottleneck of prediction with Boolean networks is the execution time. Bayesian networks are stochastic extensions of Boolean networks which allow higher discretization, are causal and have faster convergence.

A.5 Differential Equations

A.5.1 Linear Differential Equation Model

The linear differential equation (LDE) model assumes that the change of expressions of each gene over time is affected by all the other genes and computed as a weighted sum of expressions of all genes, governed by a first-order linear differential equation [109]. Here, the biological system is considered as a Markovian random walk in the space of the reacting genes. The time evolution of the system is described in a master equation having two independent variables: time t and the size n of the population of the reacting genes .

The dynamics of genes LDE model is given by :

$$\frac{\partial x_i(t)}{\partial t} = \sum_{j=1}^n w_{ij} x_j(t) + \epsilon_i \quad (\text{A.29})$$

where the partial derivatives are approximated by the differences $x_i(t + \Delta t) - x_i(t)$ in the actual implementation. The parameter w_{ij} indicates how much the level of gene j influences gene i and ϵ_i is a constant bias modeling the

activation level of a gene in the absence of any regulatory inputs. By including the parameters ϵ_i as an additional column in the weight matrix $W = \{w_{ij}\}_{n \times n}$ (A.29) can be written using matrix notation for all the genes and combining over all time points as :

$$\frac{\partial x(t)}{\partial t} = W' X \quad (\text{A.30})$$

where matrix $X = \{x_i(t)\}_{n \times T}$ and W' has the augmented vector $(\epsilon_i)_{i=1}^n$ as the last column. The weight matrix is obtained by minimizing the residual sum of square errors (RSS) when true changes in gene expression in the data is given by $\frac{\partial y(t)}{\partial t}$.

$$\begin{aligned} rss(\theta) &= \sum_{i=1}^n \left(\frac{\partial y_i(t)}{\partial t} - \frac{\partial x_i(t)}{\partial t} \right)^2 \\ &= \left(\frac{\partial y(t)}{\partial t} - \theta X \right)^T \left(\frac{\partial y(t)}{\partial t} - \theta X \right) \end{aligned} \quad (\text{A.31})$$

Differentiating with respect to W we can find the minima of the function :

$$\frac{\partial RSS(W)}{\partial \theta} = -2X^T \left(\frac{\partial y(t)}{\partial t} - \theta X \right) \quad (\text{A.32})$$

The estimate of W is given by :

$$\hat{\theta} = (X^T X)^{-1} X^T \frac{\partial y(t)}{\partial t} \quad (\text{A.33})$$

Although LDE have been successfully used to simulate small (known) biochemical pathways their use in large-scale (genome-wide) modeling has considerable limitations. This form (A.33) of linear regression over simplifies the dynamics of biological systems. It is well known that many non-linear interactions occur with cross talk and pleiotropy events. Since, microarray are limited in both quality and number of samples the DE model will be more accurate than the data itself. A BN however emphasizes on the generic principles between interacting components and hence handles cross-talk with fewer parameters.

A.5.2 Recurrent Neural Networks

Recurrent Neural networks (RNN) can describe intermediate regulation including chance events of large scale GRN [110]. Random fluctuations (noise) of biological systems are classified into external noise due to random variation of external control parameters and internal noise due to chance events in biochemical reactions. The number of nodes in the neural network is determined by the number of genes.

Similar to the LDE model the wiring define regulatory interactions between genes, which are represented by a weight matrix $W = \{w_{i,j}\}_{n \times n}$ where $w_{i,j}$ is the connectivity between gene i and gene j . A positive weight implies a stimulating effect (positive feedback) while a negative weight implies regression (negative feedback). A zero weight $w_{i,j}$ means no regulatory interaction.

A 'squashing' function transfers the regulatory input $x_i(t)$ into a normalized transcriptional response $g_i(t)$. A differential equation for accumulation of gene expression products is the difference between synthesis and degradation.

$$\frac{dx_i(t)}{dt} = s_i g_i(t) - d_i x_i(t) \quad (\text{A.34})$$

where s_i is the synthesis and d_i is degradation rate of gene i . The basal transcription is included in parameter ϵ_i .

$$x_i(t+1) = x_i(t) + h(s_i g_i(t) - d_i x_i(t)) \quad (\text{A.35})$$

where h is the step size for learning.

A.5.3 Non-Linear Differential Model: S-systems

Modeling feedbacks of regulation requires more sophisticated differential systems than the LDE model described in Section A.5.1, such as those methods that can model biochemical interactions with stochastic kinetics. The stochastic systems (or S-systems) allow feedback and are based on a series of uniform ordinary differential equations (ODE) in which the component processes are characterized by power-law functions [111], [112] as follows:

$$\frac{\partial x_i(t)}{\partial t} = \alpha_i \prod_{j=1}^n x_j(t)^{g_{ij}} - \beta_i \prod_{j=1}^n x_j(t)^{h_{ij}} \quad (\text{A.36})$$

where the first term on the R.H.S represents all the influences that decrease $x_i(t)$, while the second term represents all the influences that increase $x_i(t)$. The Parameters g_{ij} and h_{ij} are interactive effectivity with gene expressions of

gene j to gene i . Usually, the inference method tries to estimate the set of model parameters $\{\alpha_i, \beta_i, l_{i,j}, \{h_{i,j}\}_{i=1,j=1}^{n,n}\}$, minimizing the mean squared error (MSE) between experimental and estimated gene expression levels. Dynamics of gene expression levels are numerically determined from (A.36) by minimizing the MSE :

$$\text{MSE} = \sum_{i=1}^n \sum_{t=1}^T \left(\frac{\frac{d\hat{x}_i(t)}{dt} - \frac{dx_i(t)}{dt}}{\frac{dx_i(t)}{dt}} \right)^2 \quad (\text{A.37})$$

where $\hat{x}_i(t)$ is the estimated change of expressions and $x_i(t)$ is the corresponding observed value. The accuracy of estimator of parameters of interaction effectivity is low and computation time becomes very high for large networks. Hence, model selection problem is often ignored and underlying biological system is assumed to be known. In graphical models such as BN model selection is easy with use of priors and optimization methods such as GA or MCMC.

A.5.4 Non-Parametric Regression with Gibbs Prior

This method estimates GRN from gene expressions under the Bayesian framework [20], [113]. Protein complexes are modelled as virtual nodes decomposed by principal component analysis (PCA) and Gibbs priors are assumed over the structures. Following Imoto et al [21], non-parametric additive regression is used to determine the true directions of causality between interacting genes. The value $x_i(t)$ of gene expressions is given by :

$$x_i(t) = \sum_{j \in a_i} m_j x_j(t-1) + \epsilon_i \quad (\text{A.38})$$

$$\text{where } m_j x_j(t) = \sum_{p=1}^{M_{ij}} \gamma_{pj} b_{pj} x_j(t) \quad (\text{A.39})$$

where m_j are smooth spline function regression coefficient of parent j and ϵ_i is an error term having a distribution $N(0, \sigma_i^2)$ and M_{ij} is number of b-splines for gene i and parent j .

A nonparametric regression model can be written as a probability density function which minimizes residual sum of squares :

$$p(x_i(t)|a_i(t); \gamma_{a_i}; \sigma_i^2) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp \left(-\frac{(x_i(t) - \sum_{j \in a_i} \sum_{p=1}^{M_{ij}} \gamma_{pj} b_{pj} x_j(t))^2}{2\sigma_i^2} \right) \quad (\text{A.40})$$

As in the case of Boolean networks the connectivity structure s is given by a matrix $\{c_{ij}\}_{n \times n}$. To determine optimal structure of the graph we use Bayesian

learning where a prior is imposed on the parameter set θ . We use a Greedy hill climbing algorithm to find the optimal graph as minimizer of posterior using multiple starting structures s to avoid local minima. The posterior density of graph then becomes:

$$p(x|S) = \int \prod_{i=1}^n \prod_{t=1}^T p(x_i(t)|S, \theta) p(\theta|S) d\theta \quad (\text{A.41})$$

To compute the integral over parameters, $\log p(x|S, \theta)$ is expanded using the Taylor series up to the second term. The function is approximated by an unnormalized Gaussian, which is known as Laplace approximation. The Bayesian network and Nonparametric Regression Criterion (BNRC) of the graph S is defined as

$$BNRC(S) = \ln \left\| \frac{T}{2\pi} J(\hat{\theta}) \right\| - 2T * l(\hat{\theta}|S) \quad (\text{A.42})$$

where

$$l(\theta|S) = \frac{1}{T} \sum_{t=1}^T \ln p(x(t)|S, \theta) + \frac{1}{T} \ln p(\theta|S) \quad (\text{A.43})$$

$$J(\theta) = -\frac{\partial^2 \{l(\theta|x)\}}{\partial \theta \partial \theta^T} \text{ and } p(x(t)|S, \theta) = \prod_{i=1}^n p(x_i(t)|S, \theta) \quad (\text{A.44})$$

where $\hat{\theta}$ is the mode of $l(\theta|S)$. Due to complexity of the search it can only be used on small networks and for small delays.

A.6 Fusion methods

A.6.1 Naive Bayes Method

This is a probabilistic model where genes are partitioned into their separate pathways. The naive Bayesian approach has two main components: one that handles model of gene expressions and another that models the corresponding protein interaction data. The two components are then combined to provide a single unified model [25]. The simple *Naive Bayes* model presumes strong (naive) independence of gene expressions collected at different time points.

Each gene i belongs to precisely to one class or pathway denoted by $r_i \in \{k\}_{k=1}^{n_r}$ where n_r denotes the number of pathways. The naive Bayes model

defines the joint distribution of gene expressions and pathways as :

$$p(r_i = k, x_i(t)_{t=1}^T) = p(r_i = k) \prod_{t=1}^T p(x_i(t)|r_i = k) \quad (\text{A.45})$$

The pathway variable r_i is assumed to be distributed in a multinomial manner, parameterized by the vector $\theta = \{\theta_k\}_{k=1}^{n_r}$ where $p(r_i = k) = \theta_k$. Thus, each $0 \leq \theta_k \leq 1.0$ and $\sum_{k=1}^{n_r} \theta_k = 1$. The conditional probability distribution (CPD) $p(x_i(t)|r_i = k)$ are modelled using the Gaussian distribution $N(\mu_{r_i=k}(t), \sigma_{r_i=k}^2(t))$. As the interacting proteins are more likely to be in the same pathway, using a binary Markov network, we define an undirected graph with edges c , with potentials $p(r_i)$, and $p(r_i, r_j)$ which is the compatibility potential for each edge from gene i to gene j . A single parameter $\alpha \geq 1$ is used to represent:

$$p(r_i, r_j) = \begin{cases} \alpha & \text{if } r_i = r_j; \\ 1, & \text{otherwise.} \end{cases} \quad (\text{A.46})$$

The parameter α is assumed known. Increasing α results in a larger number of interactions among genes in the same pathway. The joint distribution is then defined as

$$p(x_1(t), x_2(t), \dots, x_n(t)) = \frac{1}{Z} \prod_{i=1}^n p(r_i) \prod_{[i-j] \in c} p(r_i, r_j) \quad (\text{A.47})$$

where Z is a normalizing constant. To overcome degeneracy of two models are used using the common pathway variables r_i . The distribution $p(r_i, x_i(t)_{t=1}^T)$ of gene expression can be decomposed into the potential in the interaction model. The combined model is thus a partially-directed graphical model, with $T + 1$ random variables, pathway r_i and expression times series $x(t)_{t=1}^T$

$$p(R, X|c) = \frac{1}{Z} \prod_{i=1}^n p(r_i = k) \prod_{t=1}^T p(x_i(t)|r_i = k) \left(\prod_{[i-j] \in c} p(r_i, r_j) \right) \quad (\text{A.48})$$

where c is the set of edges.

Hence, we only need to estimate the model parameters, which consists of : the multinomial θ over pathway assignments, and the means and standard deviations of each of the n_r Gaussian distributions associated with each of the $n_r.m$ CPD's $p(x_i(t)|r_i)$. A standard approach is to find the *maximum*

likelihood (ML) parameters : the parameters $\hat{\theta}$ that maximize the likelihood $p(X|\theta)$. For the pathway variables that sufficient statistics is N_{r_i} = number of genes assigned to pathway c_i . For the expression CPD $p(x_i(t)|r_i)$, the sufficient statistics are the first (mean) and second (variance) moments of the empirical distributions.

An Expectation Maximization (EM) algorithm for finding a local maximum of the likelihood function is used. Exact inference over the partially-directed graph is intractable, instead a belief propagation [114] an approximation inference algorithm which passes messages between neighboring nodes in the graph is used. The E-step uses the current estimate of the parameters to compute the distribution over the hidden variables given the observed data. Using this posterior distribution, the M-step re-estimates the model parameters using the expected sufficient statistics.

It can be seen that the Naive Bayes classifier assumes that each time point is independent of the other, however this is not true incase of time-series data. A dynamic Bayesian network is able to model these time dependencies in a Markov model.

A.6.2 Mining Cross-Graph Quasi-Cliques

Genes in a specific pathway exhibit similar expression profiles and protein products of these genes often interact. Coincidence of co-expressed genes and interacting proteins is hence biologically significant. Cross-graph mining looks for clusters in both protein-protein interaction (PPI) networks and GRN, simultaneously [26].

A complete graph is one in which all pairs of genes are connected and a sub-graph of a complete graph is called a clique if all pairs of genes are connected and no proper superset of this graph has the property. Finding cliques is a computationally expensive task. The definition to γ -complete graph was extended where each gene is connected to at least $\gamma \cdot (n - 1)$ other genes.

For a set graphs $S = \{S_i\}_{i=1}^k$, for parameters $\{\gamma_i\}_{i=1}^k$ a subset of vertices G is called a cross-graph quasi clique if $S_i(G)$ is a γ_i -quasi-complete graph for all i where $i \in \{1\}_{i=1}^k$ and there is no proper superset of S having this property. A rudimentary algorithm is to mine the complete set of γ_1 -quasi-complete graph in S_1 which have at least min genes. Next a joint-mining step checks if each subset of vertices G found is still a γ_i -quasi-complete graph in S_i . For each pathway of interest, a subset of genes G such that G is a γ_1 -quasi-clique in the protein interaction data and is a γ_2 -quasi-clique in the gene expression data,

where γ_1 and γ_2 are user-specified parameters. G is then a *cross – data* set clique.

Since both the gene expression data and protein data are very noisy, clusters obtained by PPI and GE data are more reliable. Clustering of gene expression into cliques is done using similarity measures like Euclidean distance, Pearson’s correlation coefficient etc.

A.6.3 Simulated Annealing and SVM score

This method attempts to annotate unknown genes based on GE and PPI networks [27]. The set of functional class F is obtained from MIPS functional catalogue database [115]. Functional class of gene i is denoted as r_i . Here a gene generally has multiple functions; the set of all functional classes of i will be denoted by f_i . The function-function interaction probabilities (FFI) $p(r_i, r_j)$ are estimated from the PPIN as follows :

$$p(r_i, r_j) = \frac{|FFI_{ij}|}{|PPIN|} \quad (\text{A.49})$$

where $|PPIN|$ is number of edges in PPIN and $|FFI_{ij}|$ are the number of edges where both genes i and j belong to the same functional class. It is seen that the probability of diagonals of FFI or intra-functional interactions is higher. For gene expression data, using one-against rest rule, the n_r class classification is transformed into a binary classification. The support vector machine is then used to solve the binary classification problem. For the function class r_p let the positive class precision from SVM be $p(r_p)$ and the negative class precision be $p(\hat{r}_p)$. Suppose gene i is predicted by the SVM classifier v_p with class label r_i then we use the following equation to estimate $p(r_i|v_p)$, the conditional probability of gene i ’s membership given its prediction by v_p :

$$p(r_i|v_p) = \begin{cases} p(r_p), r_i = r_p \\ 1 - p(\hat{r}_p), r_i \neq r_p \end{cases} \quad (\text{A.50})$$

Lastly simulated annealing is used to combine the above two models. The fitness of functional assignments to the expression data is measured using

$$V_{exp} = \sum_{i \in G} \sum_{r_p \in F} \log p(r_i|v_p) m_{ip} + \log(1 - p(r_i|v_p))(1 - m_{ip}) \quad (\text{A.51})$$

where the index m_{ip} indicates that gene i is assigned to class $r_p(m_{ip} = 1)$ or not ($m_{ip} = 0$) by the SVM algorithm.

Further a 10-fold CV is wrapped outside the simulated annealing to calculate the precision and recall. The method assumes interactions are likely between genes having similar function, however many inter-functional interactions are known to exist. SVM's also need a lot of data for good prediction and are very slow when the network size is large.